UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

EVANDRO MANARA MILETTO

CODES: an Interactive Novice-oriented Web-based Environment for Cooperative Musical Prototyping

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Prof. Dr. Marcelo Soares Pimenta
Advisor

Porto Alegre, November 2009

CIP – CATALOGING-IN-PUBLICATION

Miletto, Evandro Manara

CODES: an Interactive Novice-oriented Web-based Environment for Cooperative Musical Prototyping / Evandro Manara Miletto. – Porto Alegre: PPGC da UFRGS, 2009.

142 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Doutorado em Ciência da Computação, Porto Alegre, BR–RS, 2009. Advisor: Marcelo Soares Pimenta.

1. Cooperative music prototyping. 2. Networked music. 3. Interfaces for novices. I. Pimenta, Marcelo Soares. II. Título.

"To Pedro and Mara"

ACKNOWLEDGMENTS

Firstly, my special thanks to Pedro and Mara who supported me unconditionally during these long four years, for sharing all the moments, giving me help and motivation when I needed most.

All my gratitude to Professor Marcelo Pimenta for his trust and encouragement over the last years, and especially for his unconditional support, rich advices, and commitment to guiding me through my doctoral research.

I would like also to address special thanks to Professor Rosa Vicari for all the opportunities and academic experiences she has provided me.

For the inspiring ideas related with the main topics of my thesis, extremely helpful discussions, and all the support I am tremendously indebted to my friends and colleagues Luciano Vargas Flores (now at Universidade Federal do Rio Grande do Sul) and Leandro Lesqueves Costalonga (now at University of Plymouth).

My special thanks to LCM Team and collaborators with whom I've shared a rich work experience and for their collaboration, discussion, development, and help in all stages of my research project. I am indebted to several people in this regard: Professors Eduardo Miranda, Eloi Fritsch, Marcelo Johann, Damián Keller, and colleagues Aurélio Hoppe, Jérôme Rutily, Elder Santos, , Moser Fagundes, Leonardo Santagada, Marcos Hartmann, and Daniel Kuck. Special thanks also addressed to Emiliano Padilha, Michelle Leonhardt, Egon Frölich, and José Souza for their help in reviewing my text.

Also, I would like to thank Professor Jean-Paul Sansonnet, François Bouchet, and Laurent Pointal from LIMSI/CNRS for their excellent hospitality and for their help on my work during my stay in Paris.

I also acknowledge the funding support by CAPES and CNPq, which financially helped me in my stay in Porto Alegre and Paris.

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

ASDR	Attack, Decay, Sustain, Release
AJAX	Asynchronous Javascript And XML
API	Application-Program Interface
bpm	beats per minute
CME	Centro de Musica Eletronica (Electronic Music Center)
CM	Computer Music
CODES	Cooperative Music PrOtotyping DESign
CSCW	Computer Supported Cooperative Work
CMN	Common Music Notation
CMP	Cooperative Musical Prototype
CVS	Concurrent Version System
DAO	Data Access Object
DCP	Distributed Collective Practices
FMOL	F@aust Music On Line
GUI	Graphical User Interface
UI	User Interface
HTML	Hyper Text Markup Language
HCI	Human-Computer Interaction
IBIS	Issue-Based Information System
ICMA	International Computer Music Association
ICMC	International Computer Music Conference
IMN	Interconnected Musical Networks nocao
JDK	Java Development Kit
JDBC	Java Database Connector
LCM	Computer Music Lab
MIDI	Musical Instrument Digital Interface

MP	Musical Prototyping
MPR	Music Prototyping Rationale
NIME	New Interfaces for Musical Expression
NLP	Natural Language Processing
PC	Personal Computer
PSO	Public Sound Objects
QOC	Questions, Options and Criteria
RIA	Rich Internet Applications
SBCM	Brazilian Computer Music Symposium
UFRGS	Federal University of Rio Grande do Sul
CLASP	Composing, Literature, Audition, Skill acquisition and Performance
UML	Unified Modeling Language
VST	Virtual Studio Technology
W3C	World Wide Web Consortium

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Music creation is considered as mostly a solitary activity performed by musicians. Since music has traditionally served as a natural motivation for community formation, new modalities have been created by means of use of technology, and one potential convergency of social activities and music making is the field of "Networked Music". It allows people to explore the implications of interconnecting their computers, and share musical experience as a social activity through music. This thesis assumes that a networked music environment - if specifically designed for that - can stimulate social ways of music creation, even by novices in music - people assumed to have no previous knowledge of music.

The goal of this thesis is to investigate how networked music technology can provide adequate support for music creation, to discuss how it is possible to overcome a set of natural barriers, and to define requirements specifically oriented to users novices in music, and, as the main result and testbed of this research, to introduce CODES (*Cooperative Music Prototypes Design*) - a Web-based novice-oriented environment designed to support cooperative music creation.

The prototypical nature of CODES is designed and built in order to provide a novice-oriented perspective, as a novice may experiment with music by combining, listening and rearranging pre-defined sound patterns to create simple musical pieces - called *Music Prototypes*. Furthermore, CODES users may cooperate with partners in a cyclical and collaborative process of music prototypes creation - called *Cooperative Music Prototyping*, using customized awareness, argumentation, and negotiation mechanisms until a final consensual music prototype stage is reached.

Throughout this volume, the main concepts, principles, requirements and characteristics of CODES are presented, and the details of design, implementation of CODES and its evaluation by actual users are described.

Keywords: Cooperative music prototyping, networked music, interfaces for novices.

Um Ambiente Cooperativo e Interativo na Web para Prototipação Musical

RESUMO

Criação musical é considerada uma atividade individual realizada por músicos. Desde que a música tem servido como motivação natural para formação de comunidades, novas modalidades tem sido criadas por meio do uso da tecnologia e uma potencial convergência de atividades sociais e a criação musical é campo denominado *Networked Music* (música em rede), que permite que pessoas explorem as implicações da interconexão de seus computadores e compartilhem experiências musicais como atividades sociais através da música. Esta tese assume que um ambiente de música em rede, se projetado especificamente para isso, pode estimular formas sociais de criação musical, mesmo para usuários leigos em música - pessoas que não possuem conhecimento musical.

O objetivo desta tese é investigar como a tecnologia musical pode proporcionar suporte adequado para criação musical e discutir como é possível eliminar obstáculos naturais para esta atividade, bem como definir requisitos orientados especificamente para leigos em música. Como resultado principal e ambiente para teste desta pesquisa, apresentar CODES (*Cooperative Music Prototypes Design* - Projeto de Protótipos Musicais Cooperativos) - um ambiente baseado na Web, direcionado e projetado para suportar criação musical cooperativa por leigos em música.

A natureza prototípica de CODES está projetada e contruída sob uma perspectiva orientada a leigos, para proporcionar-lhes condições para experimentação com música combinando, ouvindo e rearranjando padrões sonoros para criar peças musicais simples - chamadas *Protótipos Musicais*. Ainda, os usuários de CODES podem cooperar com parceiros num processo de criação de protótipos musicais cíclico e colaborativo (chamado *Prototipação Musical Cooperativa*), usando mecanismos de percepção, argumentação e negociação personalizados para permitir que um estágio consensual final seja atingido.

Ao longo deste volume, os principais conceitos, princípios, requisitos e características de CODES são apresentados e os detalhes do projeto, implementação e avaliação com usuários reais são descritos.

Palavras-chave: prototipação musical cooperativa, música em rede, interfaces para leigos.

1 INTRODUCTION

Computer Music is a computer science discipline that includes the theory and application of (new and/or existing) technologies in music. It comprises subfields such as acoustics, sound synthesis, sound design, digital signal processing, sonic diffusion, psychoacoustics, computer aided composition, among others. Hence, computer music is an interdisciplinary and rich area for investigating many other computer science topics (MOORE 1990).

One of these topics includes cooperation over the internet that, earlier considered as an auxiliary tool for musical composition by experts (IAZZETTA; KON 1998), nowadays provides new possibilities to contemporary music, calling ordinary users' attention as well.

This cooperation tendency grows integrated with the computer music field in a new domain area recently been called Networked Music - one potential convergency of social activities and music making - subject of an special issue of Organised Sound Journal (2005). On one hand, networked music applications, as described in a survey by Barbosa (2003), associated with the Web 2.0 allow experimental artists (such as novices in music) to explore the implications of interconnecting their computers for musical purposes. It has turned the passive user into an active producer of content and shaper of the ultimate user experience. On the other hand, these networked music applications need to meet specific requirements to allow effective cooperation and musical experimentation by novices.

This thesis work emerges in this scenario: in this thesis some effort has been made to improve this user experience to go beyond the file "producing" and "sharing". A web-based environment called CODES (COoperative Musical Prototypes DESign) has been designed and developed to allow cooperative musical experimentation by novices in order to turn them *creators of musical content*.

The CODES experiments revealed several of these requirements that emerged during the tests, showing the need for adopting a broad HCI approach not only to make the CODES interaction design more efficient, but also to provide a common working method for the networked music area.

This thesis in Computer Science suggests the integration of Computer Music, Human Computer Interaction, and Computer Supported Cooperative Work fields to allow effective cooperation in musical experiments on the Web by novices in music.

This introductory chapter presents the main motivations and goals that lead up to the development of this research in section 1.1. The section 1.2 presents a summary of the contributions, section 1.3 delimits the scope of the work, and section 1.4 describes how the text of the thesis is organized.

1.1 Motivation and Goals

The Web is becoming increasingly attractive as a technology to support social activities. Today YouTube (GOOGLE 2009) and other social Web services such as MySpace (MEDIA 2009), and Flickr (YAHOO 2009) have improved the interaction between users and systems over the Web, and users are getting used to new purposes, like engagement, entertainment and self-expression. In fact, Web 2.0 has turned the passive user into an active producer of content [1] and shaper of the ultimate user experience, and the Web is becoming a rich and ideal means for social activities.

Since music has traditionally served as a natural motivation for community formation, new social modalities has been created for music listening and sharing by means of use of technology. New technologies such as weblogs, wikis, file sharing services, podcasting and social networks allow users to become an active part on the Web and let them participate in developing content (KOLBITSCH; MAURER 2006).

However, in the musical field, some peculiarities make the creation and conception processes different from those carried out in other fields. For instance, musical composition is a complex activity where there is no agreement about which activities have to be performed and in which sequence: each person (composer or not) has a unique style and a way of working. As a consequence, most composers do not have developed yet the tradition of sharing their musical ideas and collaborating while composing.

On the other hand, this thesis is focused on the web ordinary users (called "novices in music" or simply "novices" in this research), and the possibility of allowing them to participate in musical creation or experimentation.

This research investigates the possibilities of Web based networked music to allow music experimentation by novices in a cooperative way. Current related networked music works (see section 2.5) do not use the network support to explore the need for cooperation that these connected (networked) environments offer. If they are supposed to be cooperative, usually they are not designed for novices. More details about these environments can be seen in the chapter 2.

Nonetheless, there are natural barriers that need to be broken to appropriately engage novice users to experiment with cooperative musical actions. The following list enumerates these barriers (for novices) and respective problems that challenge this research. They are:

a. **Music creation**: how to allow musical creation by novices in music?

b. **Inadequate Interface**: how to provide the practice of actions of the musical technology domain (musical creation, performance, etc.) for novices ?

c. **Universal (in)accessibility**: how to provide universal access to displaced users considering technological requirements and constraints?

d. **Musical notation**: how to show understandable musical information in a high level to be used by novices, without details related to knowledge of instruments, musical notation, and musical theory?

[1] the Tofflers coined the word "prosumer" for people who are both producers and consumers of (usually internet-based) materials (TOFFLER; TOFFLER 2006)

e. **Own a musical instrument**: how to use and master a virtual instrument to produce the desired sound?

f. **Complex musical manipulation**: how to find, combine, and sequence musical samples, edit the sequences, store, and retrieve the sound?

g. **Group awareness**: how to represent, distinguish, and identify new actions and contributions from others in the context of a group? How to provide others with the understanding of your actions or musical ideas?

h. **Poor communication**: how to link users' arguments and decisions with the actions they performed to allow the refinement and evolution of an interactive musical product?

i. **Lack of negotiation**: how to provide an effective negotiation among users to get a final consensual result?

Although there exists related work in cooperative musical experimentation, as described in section 2.5, none of them is capable to solve all these questions.

Thus, one possibility herein explored is to think in musical creation as a prototyping process done by ordinary users.

Two points should be considered: a) "prototyping" is not a common expression in artistic literature and b) "composition" is a music creation activity carried out by composers. Of course, non-specialists in music are not formal composers, but they may be able to experiment with music as long as they use an adequate support. It implies new requirements that should be taken into account if we consider these novices in music as a new user profile: the Web composer (MILETTO et al. 2009). The results of their creative experiments are deliberately called *musical prototypes* in this thesis in order to highlight this difference. In the music literature, "draft" is commonly applied to such kinds of creative work, but here the emphasis is focused on the prototyping process, and not on the product itself (in which case "prototype" or "draft" correspond to the same idea).

Based on this idea, a *Musical Prototyping Process* is proposed in this thesis, and a Web system called CODES has been developed and tested with real users. CODES is a Web-based environment designed to support *Cooperative Music Prototyping* (CMP), with special focus on novices in music. The system is able to be used by ordinary users trying to experiment with music themselves or participating in a collective music creation. Aspects related to the interface, interaction, and cooperative activities are proposed, presented, and discussed in the text.

There are other aspects motivating and justifying this research. The starting point is mentioned in the editorial of the Organised Sound Journal (SCHEDEL; YOUNG 2005), as follows:

> "... Despite the existence of many persistent spheres for the creation of networked music, none seem to have yet found the combination of usability and compelling results to ignite widespread awareness and participation. There are still many questions about how best to incorporate the intrinsic characteristics of networked communication into musical form. ..."

This research attempts to address some of these questions focusing on technology issues applied to the musical field. Particularly, it aims at filling the gaps found in the related works, presented and discussed in chapter 2.

1.1.1 Thesis Goal

The thesis goal is to investigate how networked music technology can provide adequate support for music creation and to discuss how it is possible to overcome a set of natural barriers and to define requirements specifically oriented to users novices in music.

As the testbed of this research, this work introduces CODES (*Cooperative Music Prototypes Design*) - a Web-based novice-oriented environment designed to support cooperative music creation.

Thus, this work proposes new concepts and characteristics in which technology can offer great contributions to social ways of music making by novices. In fact, the main motivation of this thesis is the belief that no previous musical knowledge should be required to any user to create musical prototypes.

This thesis is focused in addressing all these aspects, presenting the answers and developing the correspondent solutions in an integrated Web-based environment to overcome those barriers, and thus allowing musical experimentation in Web communities.

CODES is designed to support effective collaboration over the Web with special focus on novices in music. Another important motivation is that non-music experts can have, through CODES, the opportunity of being, like experienced musicians, the actors of their own musical experiences. Of course, it is not a matter of musical quality of the finished work, but the mere possibility of "creating it".

Besides, the expectation of providing digital and social inclusion through the creation of a free, accessible and useful system for experimentation and entertainment also stimulates this research.

1.2 Contributions

The main contributions of this work are described as follows:

- The definition and application of the prototypical and cooperative principles to encourage novice-oriented music creation activities;
- Integration and adaptation of Computer Music, HCI and CSCW concepts, methods, and techniques. CODES uses, for example, "awareness" from CSCW and "design rationale" from HCI, which are not usually adopted by computer music researchers but help solving problems in this area;
- Definition of interface requirements for novices interested in music experimentation. For example, the high-level graphic musical representation proposed (sonic icons) through the direct manipulation (see details at section 4.4);
- Proposition of the *Cooperative musical prototyping* concept as an interactive and iterative process for collective music experimentation. This concept includes assumptions like a) novice focus is not in the quality, b) their creative actions happen without expected sequence, c) they need a tool to support

cyclic interactions with different sequences, and this creates multiple versions of a musical prototype

- Proposition of the argumentation mechanism named *music prototyping rationale* for supporting the exposition of user ideas and motives that lead users' musical actions inside the environment

- Proposition of the *versioning mechanism* aiming at providing effective negotiation and to keep the authorship of each user contribution

- Definition and construction of the CODES environment as a proof of concept.

All of these contributions of this interdisciplinary research were accepted, recognized, and published in different peer-reviewed conferences and journals of the different scientific communities, notedly for the three research fields integrated in this thesis: HCI, Computer Music, and CSCW. See Appendix A to see the published work by the author.

1.3 Scope

This thesis focuses on the technological aspects carried out during the four years of research.

The proposals and contributions presented here take into account the mechanisms, technologies, and concepts, related to architectures, interface, and interactive issues of the Computer Music, HCI and CSCW areas.

However, the following aspects are outside the scope of this thesis:

- Aesthetics and musical concerns. It does not consider the artistic and aesthetic musical results of CODES usage. This research is focused on enhancing and facilitating the musical prototyping experience.

- Scalability issues of the CODES environment. None of the non-functional tests was performed for measuring the capability of the system to scale up the number of transactions and the data volumes supported. Stress test in a Web environment such as CODES, which typically involves a huge number of users, is testing beyond the limits of normal operation, and is recommended to determine the robustness of the system. Such tests called "stress testing" was not executed so far.

1.4 Thesis Outline

This thesis is organised as follows. Chapter 2 details the main concepts and summarizes the characteristics of related work needed to understand this thesis work. It discusses the following themes: music and computer music concepts, HCI concepts, CSCW concepts, NIME, networked music, some related systems, and a final discussion at the chapter synthesis.

Chapter 3 shows an overview of the CODES environment describing its requirements and assumptions. It describes the general features, Web architectures used, general requirements taken into account, basic functionalities, and at last, a discussion about the need of assistance for novices in music.

Chapter 4 describes in a more detailed view how CODES provides support to the activities involved in the collective music creation by ordinary people (novices) over the Web, which was defined here as novice-oriented Cooperative Musical Prototyping in the Web. The requirements to be taken into account when designing interfaces for musical activities in this context are also pointed out. Moreover, this chapter presents the details of the interface and interaction issues taking in account those requirements, and hence the cooperative activities offered by the system.

Chapter 5 presents and discusses the tests and evaluation carried out in real contexts with real users. The tests have followed well-known subjective evaluation methods from the HCI field, in order to obtain qualitative and quantitative results by using CODES environment.

Chapter 6 discusses the main conclusions, and includes a summary of the contributions and perspectives for future work.

Appendix A enumerates the paper written during the thesis development. Appendix B shows the forms with the questions used during the CODES tests. Appendix C contains some illustration produced during the development process of CODES . Appendix D presents the main scenarios related to CODES usage, and Appendix E presents an extended abstract of this research in portuguese.

2 FUNDAMENTALS AND RELATED WORK

This chapter describes a set of concepts and related work aiming to contextualize the fundamentals and the main issues and contributions of this thesis as an interdisciplinary endeavor.

Its purpose is to create a terminological and conceptual grounding to comprehend the needs of novice users and also to establish a conceptual framework to understand the proposed solutions presented in chapters 3 and 4. It is organised as follows: section 2.1 presents the music and computer music concepts used here to inspire our proposals regarding to sonic perspective. Section 2.2 presents the concepts and features related to interface and interaction aspects of the work. Section 2.3 presents CSCW issues aiming at discussing the collaboration support. Section 2.4 presents an attempt of merging computer music with CSCW. At last, section 2.6 synthesizes, discusses and concludes the remarks of this chpater.

2.1 Music and Computer Music Concepts

The use of computers in music has opened up new possibilities for amateur and professional musicians alike. Research in the field of Computer Music is directed towards the construction of computer systems for composition, performance of musical tasks, music training, signal processing and extension of traditional music sounds, notation study, music analysis, storage and communication of musical data, and music information retrieval and classification, among others.

This thesis is focused on the support for the musical prototyping activities for non-especialists in music. This is a kind of experimentation in which users can use sound samples in sequence to experiment with "music creation", inspired on the concepts in the next sections.

2.1.1 Musical Composition

Composition is the process of creating an original piece of music (ISAACS; MARTIN 1985). A composer is anyone who practices composition by using compositional techniques and methods to create music.

Composition comprises musical elements, which may vary widely between cultures and from composer to composer. Despite most composers affirming that initial inspiration is essential, a previous knowledge about composition techniques is required.

It is possible to compose music without inspiration. There is a lot of possibilities to solve musical questions by following some techniques and compositional rules

(SCHOENBERG 2008).

There exists a large corpus of formal knowledge associated to musical composition and such knowledge, in general, discourages the composition by novices.

However, composers and musicians claim it is not possible to compose music without musical knowledge and without putting into practice some compositional techniques and rules. In fact, there is no available literature about novice-composed music or even music created by lay people.

2.1.2 Musical Performance

In performing arts, a performance generally comprises events in which a performer or a group behaves for another group of people (the audience).

In the case of music, this is the step in the musical process during which musical ideas are realized and transmitted to a listener.

A performer can determine aspects of any music he plays. Issues of tempo, phrasing, dynamics, and, in some kinds of music, pitch and instrumentation are subject to a performer's discretion.

Examples range from singing solo, in a choral or in a band, performing in a ballet, playing turntable and computer music systems, among others.

Music is eventually performed by lay people in a context of bands as amateur music groups. In fact, for lay people to perform music is necessary to overcome some natural barriers such as described in section 1.1. Besides, performing techniques and music knowledge are also required for musical performance.

2.1.3 Computer-aided Musical Composition

Interests in musical composition by computers date back to the 1950's when Markov chains were used to generate melodies (FRANKLIN 2006). Computer assisted composition or computer-aided composition is part of the general field of computer music. It is the technique or practice of using a computer to aid in the musical composition, though the music itself may be performed either electronically or on traditional, non-electronic instruments without the use of a computer or any kind of electronic device.

There exist a significant literature about this theme since the previous decade. Some surveys address the use of computers in music composition (LOY; ABBOTT 1985; LOY 1989). In the book "Composing Music with Computers" (MIRANDA 2001), the author focuses on the role of the computer as a generative tool for music composition. He discusses a number of computer music composition techniques ranging from probabilities, formal grammars and fractals, to genetic algorithms, cellular automata and neural computation. "Electronic and Experimental Music" (HOLMES 2002) is an introduction both to the theories of electronic sound and sound production and to the history of some of the earliest experiments in instrument building and composition.

The wide variety of processes involved in composition and the vague and abstract nature of high-level musical material both place special requirements on the design of software tools to support composers.

2.1.4 Music Representation

Music offers a challenging array of representation problems. Music can contain symbolic or structural relationships existing within and between the dimensions of pitch, time, timbre, harmony, tempo, rhythm, phrasing, and articulation (DANNENBERG 1993). The representation of musical time, for example, has been a topic of numerous proposals in music representation research (ROADS 1996; HONING 2001).

The establishment of the Musical Instrument Digital Interface - MIDI, in the late 1980s allowed hobbyists and musicians to experiment with sound control in ways that previously had been possible only in research studios. MIDI is the most prevalent representation of music, but what it represents is based on hardware control protocols for sound synthesis. Programs that support sound input for graphics output necessarily span a gamut of representational categories.

MIDI and many other existing proposals are presented in "Beyond MIDI: The HandBook of Musical Codes" (SELFRIDGE-FIELD 1997), which is a standard reference work, describing a vast number of approaches to the representation of musical information for purposes of computer processing.

Several initiatives have addressed the need for a standardized markup-based music notation. Some examples are MML[1] - Music Markup Language, and MusicXML[2]. (CoverPages[3] and Music-notation.info[4] can be accessed for complete list about markup music notation).

Despite the great number of approaches in music representation oriented to expert people, an insignificant number is dedicated to lay people in music. Systems designed for novices require special attention both in musical interfaces as in musical representation issues.

2.2 HCI Concepts

Human-Computer Interaction - HCI, also known as Computer-Human Interaction - CHI[5], is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use, and with the study of major phenomena surrounding them.

HCI is a very broad discipline that encompasses different specialties with different concerns regarding computer development such as computer science, sociology, anthropology, ergonomics, linguistics, and industrial design.

This section presents the "usability", "accessibility", "interaction design", and "design rationale" concepts originally emerged from HCI and usually put into practice in HCI work.

The expectation is to use such concepts to make the user experience more useful and usable, and to increase their level of satisfaction, reducing critical errors.

---

[1] http://www.musicmarkup.info/index.html

[2] http://www.musicxml.org/xml.html

[3] http://xml.coverpages.org/xmlMusic.html

[4] http://www.music-notation.info/en/compmus/notationformats.html

[5] See the ACM Special Interest Group on Computer-Human Interaction at http://sigchi.org/cdg/index.html

2.2.1 Usability and Acessibility

Usability is a qualitative attribute that assesses how easy it is for users to interact with interfaces using the system functionalities properly (NIELSEN 1997).

According to Shneiderman (2004), to evaluate the system usability, experts should focus on the following usability measures:

1. **Time to learn**. Also known as learnability, it means how long does it take for typical members of the user community to learn how to use the actions relevant to a set of tasks.

2. **Speed of performance**. How long does it take to carry out the benchmark tasks?

3. **Rate of errors by users**. How many and what kind of errors do people make in carrying out the benchmark tasks?

4. **Retention over time**. Also known as memorability. How well do users maintain their knowledge after an hour, a day, or a week?

5. **Subjective satisfaction**. How much did users like using various aspects of the interface?

In entertainment applications, ease of learning, low error rates, and subjective satisfaction are paramount because use is frequently discretionary and competition is fierce (SHNEIDERMAN; PLAISANT 2004). Users will abandon the use of the system or try other if they cannot succeed quickly.

Accessibility is a general term applied to describe the degree to which a product (e.g., device, service, environment) is accessible by as many people as possible. The term is strongly related to universal design when the approach involves direct access. The Web Accessibility Initiative (WAI[6]) develops strategies, guidelines, and resources to help make the Web accessible to people with disabilities. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web (W3C 2009). Also, when designing for ordinary users, some simple rules should be considered to make a Web site more accessible. They are: a) keep the content and design of the Web site as simple as possible; b) ensure that all Web site navigation is in one location, and c) use illustrations, icons, and other visual tools to communicate key ideas.

These are some of the rules intended to be applied in this research to allow that any ordinary user using any kind of Web browsing technology must be able to visit CODES and get a full and complete understanding of the information contained there, as well as have the full and complete ability to interact with it.

Both usability and accessibility are key concepts when it comes to Web development. By using them, direct access and users' abilities to find information and satisfaction with Web sites improve significantly.

[6]See more details at http://www.w3.org/WAI/

2.2.2 Interaction Design

This discipline involves the design of how a user communicates or interacts with a computer and defines the behavior of products and systems that a user can interact with. Interaction designers focus on the flow of interaction, the dialogue between person and computer, how input relates to output, stimulus-response compatibility, and feedback mechanisms (PREECE; SHARP; ROGERS 2007).

Certain basic principles of cognitive psychology provide grounding for interaction design. These include mental models, mapping, interface metaphors, and affordances. Many of these are laid out in Donald Norman's influential book "The Design of Everyday Things" (2002).

Academic research in Human Computer Interaction (HCI) includes methods for describing and testing the usability of interacting with an interface, such as cognitive dimensions and the cognitive walkthrough.

2.2.3 Design Rationale

Design Rationale (DR) is the ability for linking argumentations to steps of a design process, proposed originally in the Human-Computer Interaction area (SHUM 1996). It is a communication mechanism among design team members, to communicate past critical decisions, which alternatives were investigated, and the reasons behind the chosen alternative.

The use of a design rationale system can improve dependency management, collaboration, reuse, maintenance, learning, and documentation.

There are many models and notations for DR, like Issues Based Information System - IBIS (RAMESH; DHAR 1992), and Questions Options and Criteria - QOC, see (SHUM 1996) for a good summary.

Nowadays, DR is adopted also by other disciplines (e.g. Requirements Engineering) and recognized as a possible way to allow a group member to obtain a better understanding of other group members' actions and decisions.

Design Rationale also helps designers to avoid the same mistakes made in the previous design. This can also be helpful to prevent duplication of work (BURGE; BROWN 2000).

Even DR has been created for documenting the reasons and motivations of the design, nowadays their main notations have been used for argumenting systems too. Such systems foster the negotiation process and allow users to position themselves in a context of a group.

2.3 CSCW Concepts

The term computer supported cooperative work (CSCW) was first coined by Greif and Cashman in 1984, at a workshop attended by individuals interested in using technology to support people in their work (GRUDIN 1994). According to Carstensen and Schmidt (1999), CSCW addresses "how collaborative activities and their coordination can be supported by means of computer systems".

Despite the name, this field of study doesn't restrict itself to issues of "cooperation" or "work", but may examine competition, socialization, and play. The field typically involves anyone interested in software design and social and organizational behavior, including business people, computer scientists, organizational

psychologists, communications researchers, and anthropologists, among many other specialties (DESIGN 2009).

2.3.1 3C Collaboration Model

This field of study examines how technology affects group interaction, collaboration, and how technology can be best designed and built to facilitate group work.

During the interactions, members of a group need to communicate with each other, organize themselves, and cooperate in a shared workspace. According to Ellis et al (1991) these three dimensions are the base of "3C Collaboration Model" represented in figure 2.1.

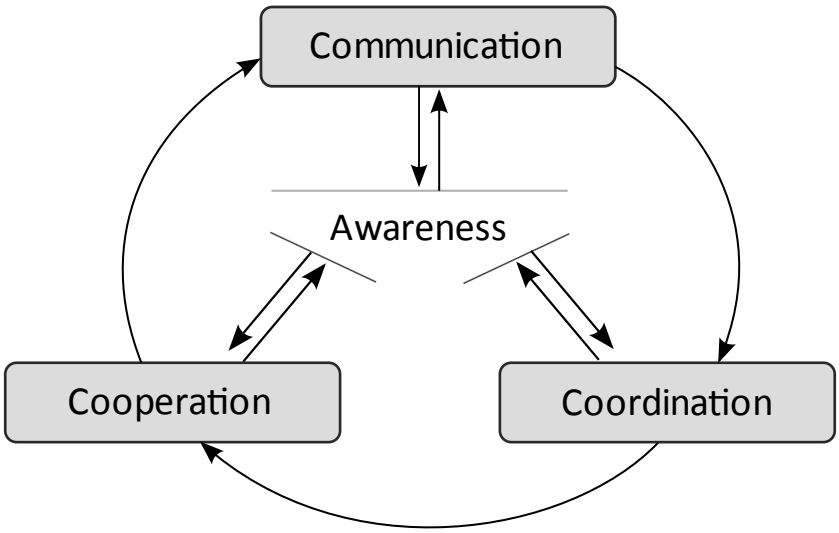


Figure 2.1: A 3C collaboration model

Concerning the collaboration, the "communication" is related to the action: negotiations are carried out, decisions are made, deals and goals are established (FUKS; GEROSA; PIMENTEL 2003).

The "coordination" of a group's activities aims at organizing members in order to accomplish the deals from the negotiation stage (RAPOSO et al. 2001). "Cooperation" is the joint production of members of a group within a shared space, generating and manipulating cooperation objects in order to complete tasks (FUKS et al. 2005). The model shown in figure 2.1 denotes the iterative aspect of the collaboration. During this iteration, information exchange is done by generating new appointments and tasks to be performed. Most of the time these tasks are managed by the coordination using the communication mechanisms to negotiate and decision making, which might generate new appointments and start a new iterative cycle.

CSCW systems can be conceptualized according to the context of a system's use. One such conceptualization is the CSCW Matrix (see figure 2.2), which appears in (BAECKER et al. 1995). The matrix considers work contexts along two dimensions: first, whether collaboration is co-located or geographically distributed, and second, whether individuals collaborate synchronously (at the same time) or asynchronously (not depending on others to be around at the same time).

Many software designed to group assistance fail not due to technical problems, but because social and human aspects related to user interaction were not considered.

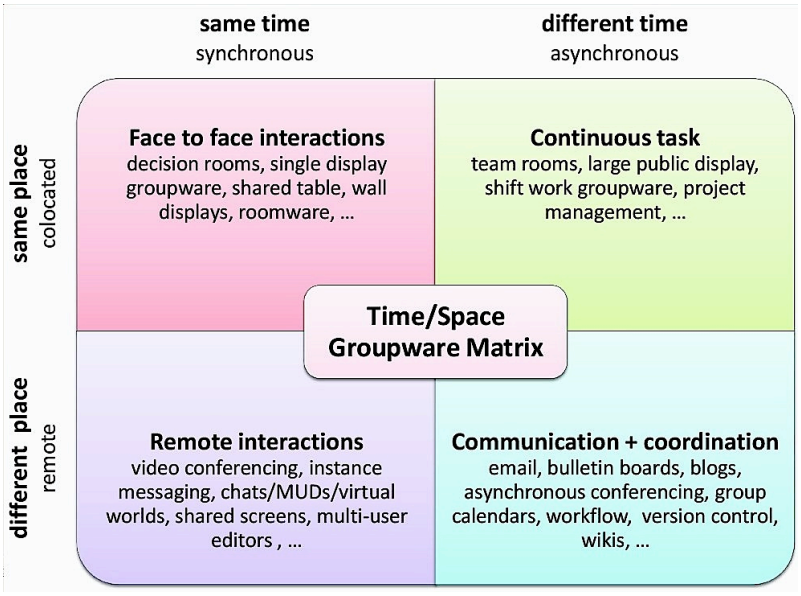


Figure 2.2: The CSCW Matrix

The great challenge in this case is to design a system with clear tasks, goals, and with adequate interaction design to the public-target.

Such challenge has encouraged this research towards the definition of and adequate support for a specific public-target like those in CODES: the novices in music.

2.3.2 Awareness

One of the most critical issues in CSCW is to have adequate support of the user's actions. Awareness is the other dimension inside the 3C model (see Figure 2.1) that stands for acquiring information, by means of the senses, about how the work of the group is being developed (FUKS; RAPOSO; GEROSA 2002). Users need to know and get feedback about actions performed in the shared workspace. It allows users to evaluate their work in the group, redirect activities if necessary, and predict future situations.

According to Borges et al. (1995) the context is an essential aspect of CSCW, besides the individual contributions, and its goal and meaning for the whole group. Awareness is to provide this context to the group members.

In awareness, support tools allow to identify a set of important features about five basic questions (4W1H):

1. What? (related to activities and roles)
2. When? (related to events, presentation, persistency)
3. Where? (related to space and metaphor)
4. How? (related to interface and the balancing - filters, group)
5. Who? (related to author of an action, presence and communication tools)

Depending on the nature of the tool and its localization in the matrix of figure 2.2, each one of these questions identify crucial aspects inside the cooperative or groupware tool.

2.4 Networked Music

According to Barbosa (2006) "Networked Music" was mentioned by Roger Dannenberg in his Keynote Speech from International Computer Music Conference 2003, by referring to it as a promising research topic.

The Cambridge Press' Organised Sound Journal has also dedicated a special issue to Networked Music in 2005 due to the relevance of the topic.

Jason Freeman has defined in his speech at ICMC 2005 as follows:

> "It is about music practice situations where traditional aural and visual connections between participants are augmented, mediated or replaced by ellectronically-controlled connections."

Networked music integrates in a certain level the Computer Music and CSCW fields.

Based on this idea, Alvaro Barbosa (2006) proposes some networked music system categories using as a reference the classification criteria of CSCW presented in figure 2.2. The graphical representation is analogous to Tom Rodden's Classification Space Dimension (RODDEN 1991), representing Networked Music Systems as shown in figure 2.3.

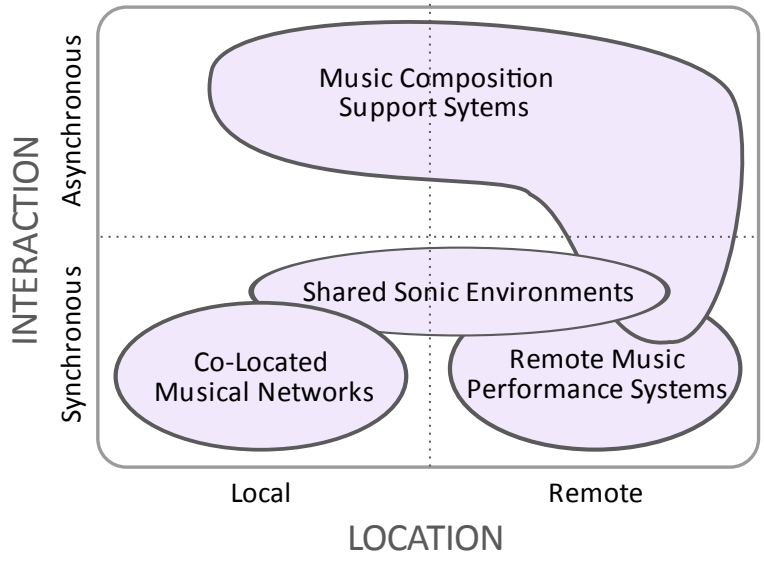


Figure 2.3: A Classification Space Networked Music Systems

1. **Co-Located Musical Networks** - used for performances with interaction in real-time in the same physical location provided by a fast local computer network.

2. **Music Composition Support Systems** - used for music composition and production. It allows geographical displacement and synchronous collaboration.

3. **Remote Music Performance System** - used for groups of remote performers/users, displaced in space, interacting synchronously.

4. **Shared Sonic Environments** - explores the distributed and shared nature of the internet, suitable for synchronous improvistation, do not requiring musical skills from the participants.

These are not closed categories according to the author, and some applications could belong to different classes if considered as a less wide ranging classification criteria.

Current networked music systems use different approaches to different target publics. User interface, network technology, musical representation, and collaboration aspects are the main challenges to the system designers.

In fact, no dominant technique or approach has yet emerged to represent what is unique in the networked music paradigm.

Besides, this thesis also suggests another important characteristic not mentioned in the classification space proposed in (BARBOSA 2006). It lacks the Orientation dimension, which means the public-target and defines if the system is addressed for "musicians" or "novices in music". It can be considered as an extension of such classification space and is adopted here in the context of this thesis.

Figure 2.4 presents this new dimension inserted in the Barbosa's Networked Music Classification Space.

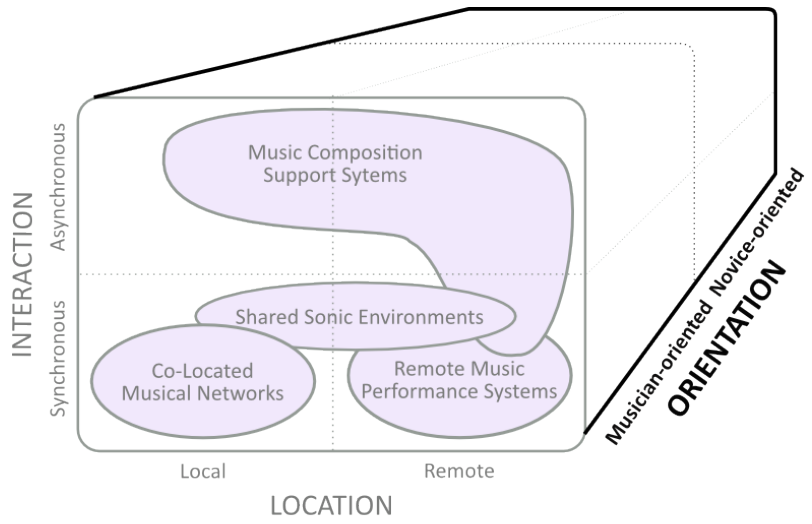


Figure 2.4: Adaptation of the Classification Space Networked Music Systems

Musician-oriented music systems usually include full and complex information, concepts, and functionalities in the interface which is part of the "musician's world", such as musical notation, filters, oscillators, among others, which ordinary users do not understand. Such systems are used by composers, musicians, and performers for musical composition, musical performance, and sound synthesis purposes.

Novice-oriented music systems should include high-level musical information to allow ordinary users manipulate musical concepts easily and transparently. Such

systems are designed for ordinary users to experiment with music in creating sequences of musical samples as well as using some kind of control (check, uncheck, draw, drag and drop, etc.) to generate some sound result. The difference is in the nature of the process: that is merely exploratory in the case of novices in music.

This research is focused specially on this target public, the novices in music, and the questions surrounding their interaction with Web interfaces for musical creation purposes. More considerations about interfaces for novices and musicians are presented in section 3.3.

The next section aims at reviewing the most representative systems which are concerned with the concepts mentioned so far, and at focusing on musical experimentation by novices.

2.5 Related work

This section summarizes the characteristics of the main novice-oriented environments found in literature for collective musical creation or experimentation on the Web. Their main features are presented, compared and discussed here.

2.5.1 Daisyphone

Daisyphone[7] is an environment for remote group music improvisation in which a user can co-create short loops of music with friends in real-time by pressing little dots to create and remove notes (BRYAN-KINNS; HEALEY 2004). The application focuses on the representation of looping music and provides support for remote collaboration.

Daisyphone's interface is made up of four main elements: 1) representation of the musical loop itself which takes up most of the screen real estate: players click on the circles to set and unset notes which are played as the rotating grey arm passes over them; 2) the modal control of the player's instrument and volume in the centre of the Daisyphone; 3) the session selector in the top left hand corner of the Daisyphone, and 4) the continual annotation over the whole interface. See a screenshot in the figure 2.5. The most striking reflection on this project is the short

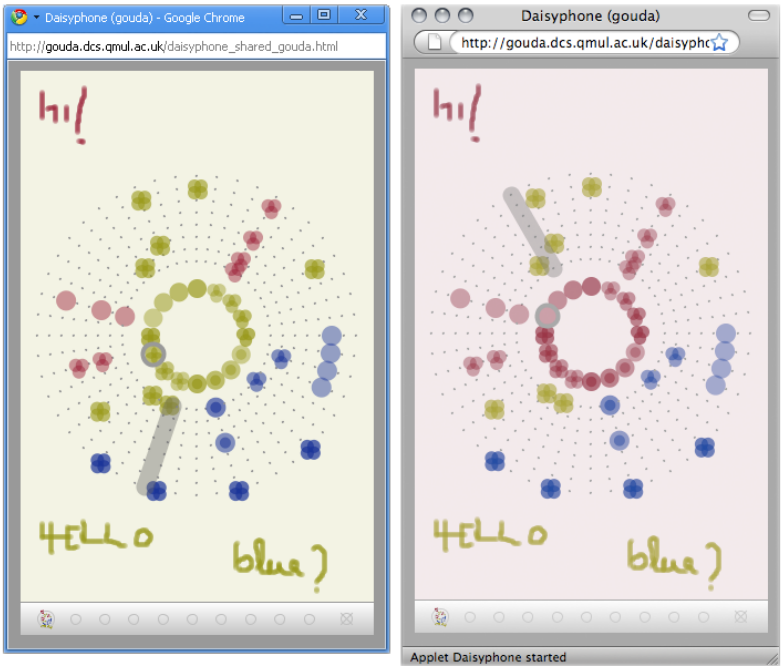


Figure 2.5: Two users interacting with Daisyphone

amount of time players had to learn to use Daisyphone, according to the author. Another interesting aspect of the observations of Daisyphone's use is the frequent writing of player's names which happened more in the public version of Daisyphone. The author suggests that this indicates that there is a need for some way in which players "make it mine" and publicly indicate their ownership of spaces over and above being assigned a specific color.

[7] http://www.daisyphone.net/

2.5.2 PitchWeb

PitchWeb (DUCKWORTH 1999) is described as a virtual instrument which is played by selecting and manipulating shapes which are mapped to sound samples. It is part of the Cathedral Project[8], which is an interactive work of art and music that allows musical composition, and is designed for the World Wide Web.

PitchWeb allows users to play music alone, or with other users over the Web for a live, real-time interactive performance.

The GUI has a palette with 64 different colored shapes, each of which is mapped to a specific sound, which can be heard when the cursor is moved over it. The shapes can be dragged onto the main playing area to form a sound score. The shapes can also be resized, as can be seen in figure 2.6, which will change how they sound when played. In this case, when the scene is played, the modified shapes play its

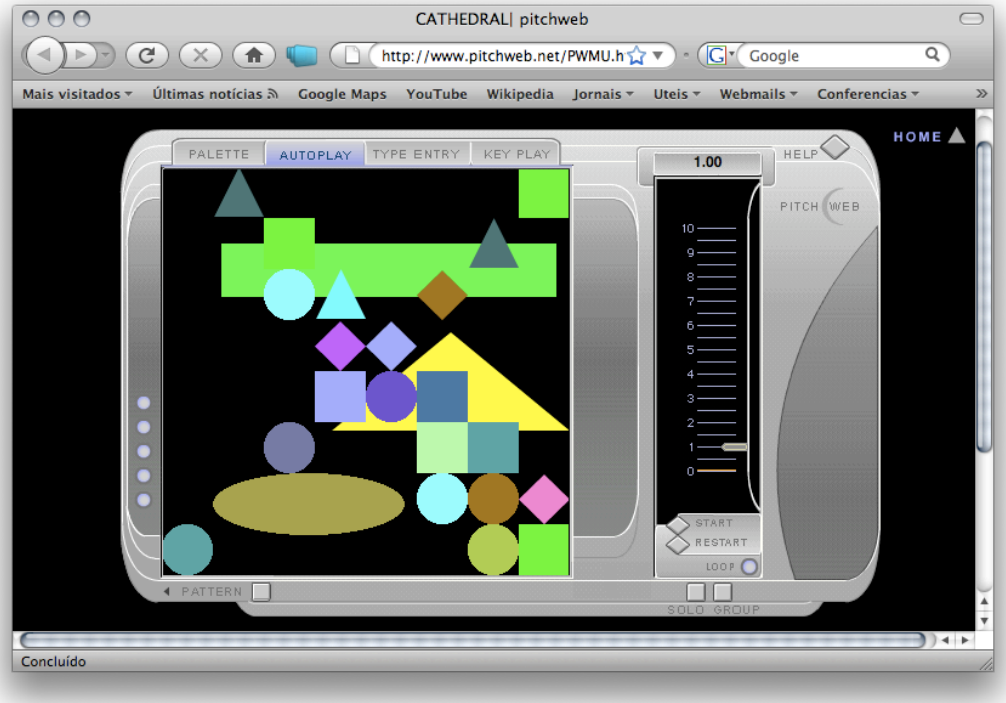


Figure 2.6: Screenshot of PitchWeb

corresponding sound for a shorter or bigger duration than the original ones.

The entire score can be played automatically by choosing a pattern of play (linear, spiral, zigzag, diagonal), which determines the pattern in which a moving dot crosses the screen, touches the shapes, and generates sounds.

Users interact in a private or solo mode or synchronously in a group, talking to each other by means of a chat tool. PitchWeb has a status area in which users can get to the actions performed in the environment.

The application was developed with Macromedia Shockwave technology and requires a Quicktime plug-in to run.

[8]http://www.monroestreet.com/Cathedral/main.html

2.5.3 WebDrum

The WebDrum[9] (BURK 2009) is Web-based application inspired on a traditional drum pattern editor where users turn on or off notes on a grid. Synthesized drum sounds are buffered in a time-stamping event and used in order to avoid downloading large audio sample files, which allows a precise synchronization. Web users can play and listen to other participants' edits and add their instrument sounds to their own pallets.

To collaborate with other online user it is necessary to get into the same room. After loading the main interface, the user clicks the "Own" button to gain control of a drum that someone else owns to edit that drum. To turn on or off notes in a melody, a user should click on a column in the melody grid (see figure 2.7).

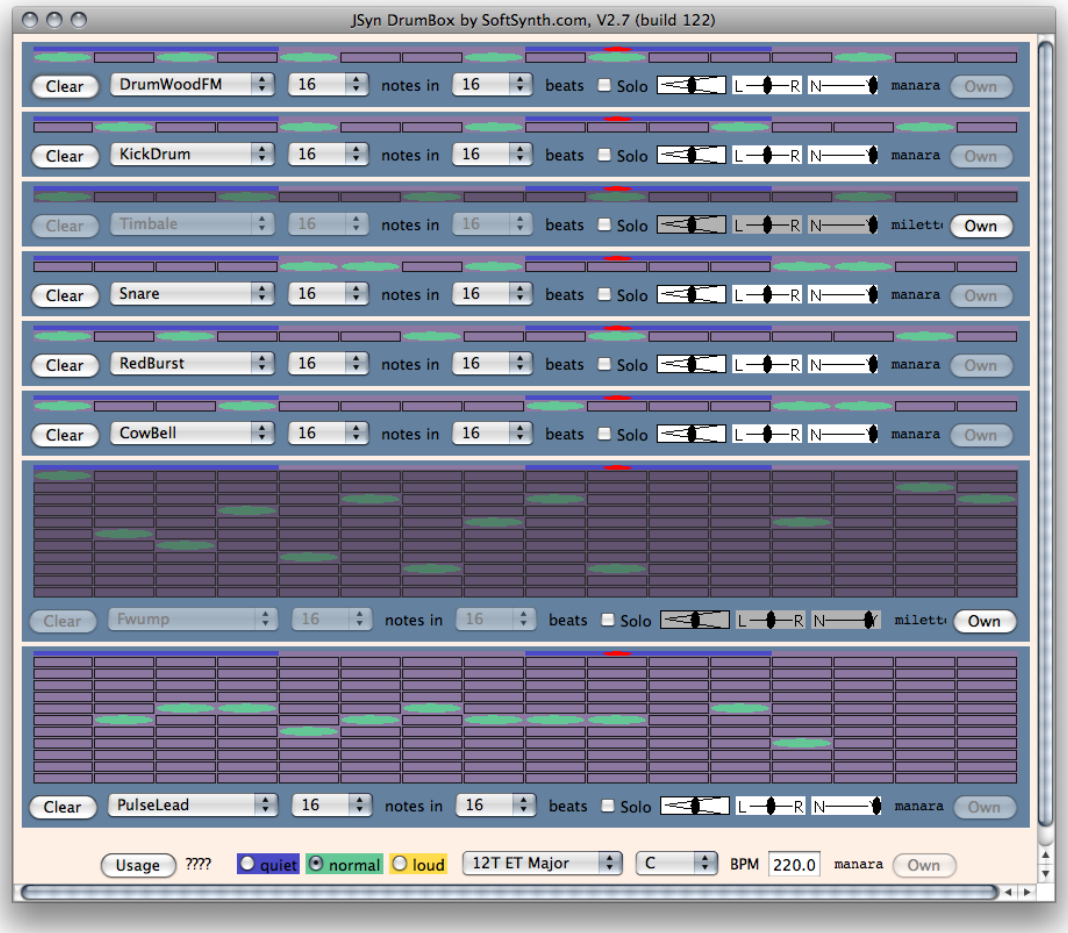


Figure 2.7: Screenshot of WebDrum on the Web

Only one note per column may be selected. To turn off a note, it is necessary to click on it.

WebDrum uses a kind of timeline (a vertical red line, formed by red points) to give a feedback to users about which notes are playing.

Users have available also a chat window to chat with other performers in a synchronous way. The chat session is not saved when users leave the application.

[9]http://www.transjam.com/webdrum/webdrum.html

2.5.4 Public Sound Objects

The Public Sound Objects - PSOs(BARBOSA 2005), is a project that consists of the development of a networked musical system, which is an experimental framework to implement and test new concepts for online music communication. The PSOs project[10] approaches the idea of collaborative musical performances over the Internet aimed at going beyond the concept of using computer networks as a channel to connect performing spaces. This is achieved by exploring the internet's shared nature in order to provide a public musical space where anonymous users can meet and be found performing in collective Sonic Art pieces.

The system itself is an interface decoupled Musical Instrument, in which a remote user interface and a sound processing engine reside with different hosts in an extreme scenario where a user can access the synthesizer from any place in the world using the World Wide Web.

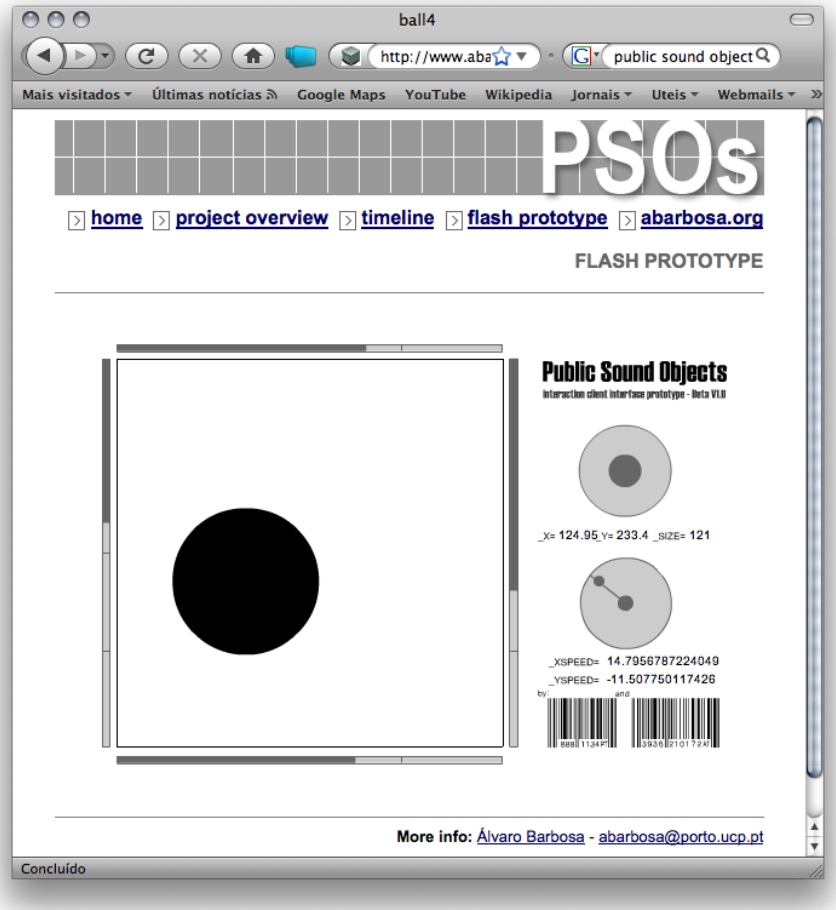


Figure 2.8: Screenshot of the Public Sound Objetcts in the Web

Specific software features were implemented in order to reduce the disruptive effects of network latency, such as, dynamic adaptation of the musical tempo to communication latency measured in real-time, and consistent sound panning with the object's behaviour at the graphical user interface.

[10]http://www.abarbosa.org/pso/psoflash/index.html

2.5.5 EduMusical

The EduMusical system (BENINI et al. 2004) supports collaborative and interactive distance learning, aiming at teaching music to children and teenagers, oriented by music instructors from an actual orchestra - OSESP, the Symphonic Orchestra of São Paulo. Collective composition is possible through the interaction among students in virtual classrooms over the Web[11], guided by a tutor.

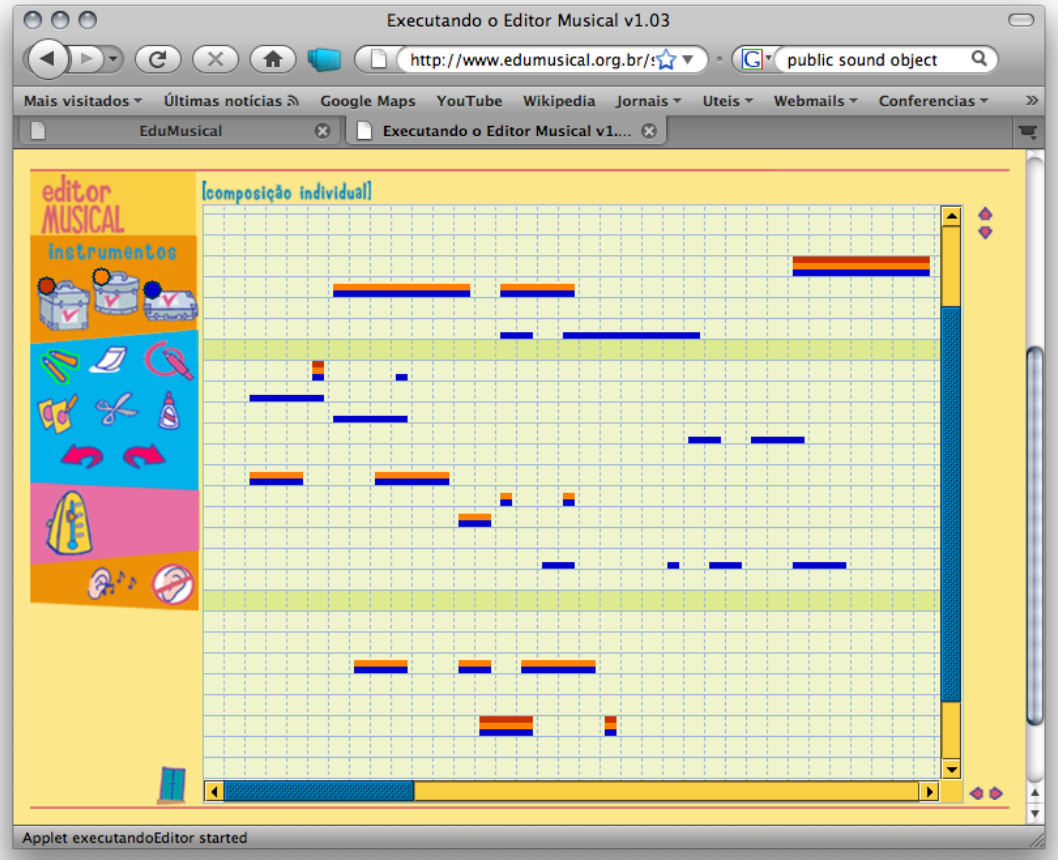


Figure 2.9: The Edumusical System

Collective composition is done through the interactions between students in the virtual classrooms. Learning in EduMusical can be individual or collective (using the virtual classrooms in this case). The result of the students' interactions is the collective and synchronous musical composition.

It is also possible to edit a musical composition remotely, in an asynchronous way. In this case, the shared composition in the server-side is updated when the users connects again in the virtual classroom.

The users communicate with each other by chat window. The collaborations happen via a "collaborative mouse", which can be owned by one user. This user selects part of the composition that is viewed by other users.

The system uses JavaSound library and an JavaApplet running in the client browser.

[11] http://www.edumusical.org.br/

2.5.6 JamSpace

JamSapce is an interactive music environment to support real-time jamming by novices and amateur musicians over a network (GUREVICH 2006).

The system takes advantage of the low latency and connectivity of a local area network (LAN) to allow real-time rhythmic collaboration from isolated locations. Two important features provided by the technology of a large LAN are leveraged in the design of JamSpace, in order to suit novice musicians. These are 1) low latency, and 2) connectivity in isolated locations. The design of JamSpace makes use of these features to offer constrained, real-time rhythmic performance with a user interface that maintains privacy and anonymity.

The JamSpace software uses a client-server architecture, in which the client GUI consists of five components: a scratch track for the local user, a set of tracks from the JamSpace, an interface for making connections to the server, tempo and metronome settings, and a matrix for managing real-time jams with other clients.

See an excerpt of the JamSpace client GUI in Figure 2.10.

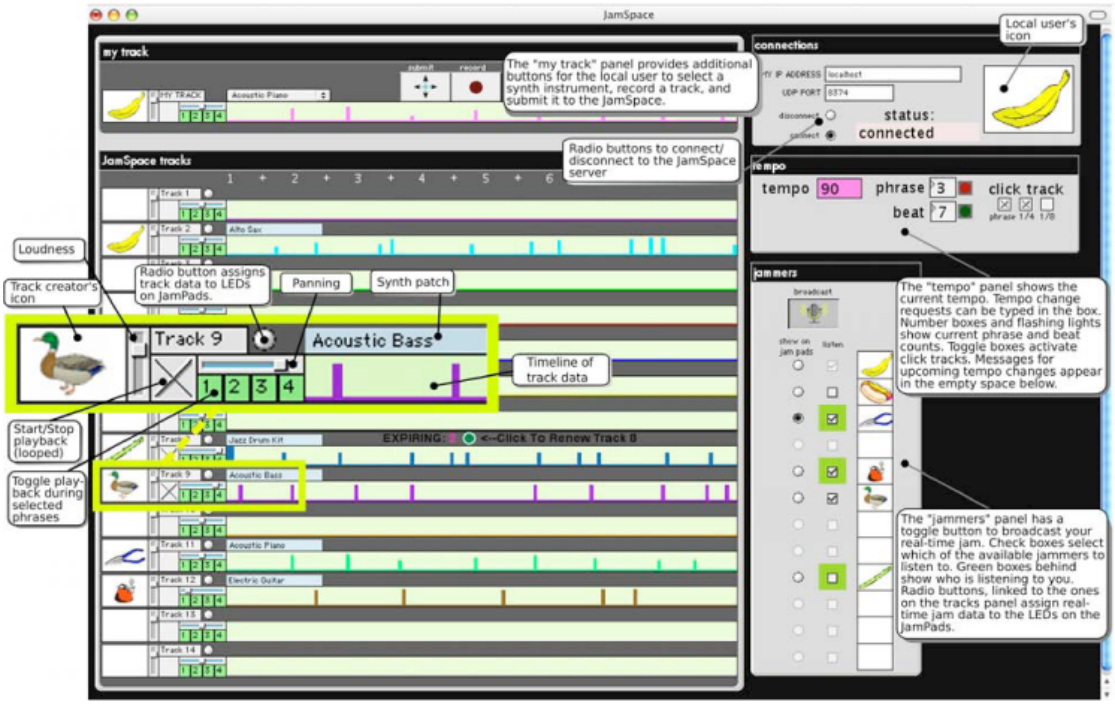


Figure 2.10: JamSpace client GUI

JamSpace's design also addresses some needs like the preservation of anonymity and privacy for amateur musicians in a group setting, with a novel hardware and software interface incorporating listening, private rehearsal, mixing, looping tracks and real-time jamming.

According to the author, the overall design philosophy of JamSpace was to begin with a specific technological platform (local network) and application area (recreational music), and then leverage their affordances to find novel interactions that address the requirements of the scenario.

2.6 Analyzing the Related Systems

This chapter has presented a review of concepts and technologies related to the thesis' main subject. This collection of related work is intended as a snapshot of the state of novice-oriented networked music applications on the Web.

In the next section, the criteria used for comparison and presented in section 2.6.2 are described.

2.6.1 Criteria for comparison

In the development process of a networked music several aspects must be taken into account.

Below, sixteen criteria are presented as being significant in order to provide flexibility and the minimum requirement for use by novices in music.

Considering the interdisciplinary aspect of this research presented in this chapter, they were divided in their respective disciplines and are the following:

Technological and architectural criteria

- **Architecture**: the architecture must define the rules, guidelines or constraints, systems structure, technical framework, and product technologies for creating conformant implementations of the system. It allow to know how to implement the core of the system and functionalities to attend the requirements as well as to know their limitations.

- **Persistence**: it is directly related with the previous criteria, which is the characteristic of state that outlives the process that created it. It means, in other words, the capacity of the system of storing the state of the product development in order to allow iterative work.

Computer music criteria

- **Sound format**: concerns the type of sound file. In general, MIDI, wave, and MP3 are the most used. However, it implies both the technological and the aesthetic aspects, in terms of file manipulation and sound result.

- **Musical process**: defines what kind of musical activity the system will provide support to. According to the concepts of Music and Computer Music presented in section 2.1 (composition, performance, synthesis, etc.) it will imply different system requirements, regarding the process and target-public.

- **Music metaphor**: it was considered a very important issue by the viewpoint of the user interface. Different musical metaphor are proposed in the literature and most of them are musician oriented. The level of concepts and musical representation used to express the musical objects will define the requirements of use and the system orientation as well.

- **Sound exportation**: is related to the capacity of exporting sounds in different formats in order to be compatible with other software. Compatibility with other systems is important to many users who may want to exchange files with other user or studios.

HCI criteria

- **Accessibility**: here, the system accessibility is more related to the restrictions that could interfere with the ability of user to access the system. Specific versions (e.g. plug-ins), hardware or specific network architecture could be some examples of this restriction of access, because ordinary users tend to avoid the execution of operations that they do not know.

- **Interaction**: this criterion reveals the implications for the system design regarding the synchronous and asynchronous interaction. Both of them presents advantages and problems. Flexibility (any time and any place), time to work, and cost-effective are, for example, related to synchronous interaction. On the other hand, real time interaction and quick feedback are, for example, advantages of synchronous interaction. There, however, different requirements and need of implementing one and other. Naturally, synchronous interaction demands a more complex control and management of realtime actions.

- **Requirements to run**: the importance of this criterion is related with the easiness of using and playing the system. When considering a target public of novices, additional software installation, specific platforms or system configurations should be avoided as much as possible.

- **Target public**: this criterion is essential since it will affect several other criteria. The less specialized is the audience, the more flexible should be the system to support them. Otherwise, people get confused and cannot be able to use the system.

- **Interaction trace**: this criterion means the traceability of the user and their actions in the system. It will provide the verification of the history or application of an item or action by means recorded identification.

CSCW criteria

- **Communication tools**: as part of social and collaborative software, communication tools encompasses a range of systems that allow users to interact exchanging or sharing ideas. Communication tools typically handle the capturing, storing and presentation of communication, usually written but increasingly including audio and video as well. They focus on establishing and maintaining a connection among users, facilitating the mechanics of conversation and talk.

- **Group memory**: in collaborative systems, this criteria plays an important role on the creative process since it usually causes effects on individual idea generation. The main goal of a group memory in this context is to capture the knowledge and accumulated contributions of the group and makes them available to others.

- **Awareness**: this criterion will provide to users the context of the actions and activities inside the environment. As users interact with the system and with other users, marks and symbols are generated aiming to inform them about such actions performed.

- **Authorship**: it means the capacity of a collaborative system of identifying and register the authors' contributions. Even if the main author of the project is the group itself rather than a single person, the fact of keeping the authorship of contributions in successive versions of a product will help with the understanding about the users and each contributions towards a final version. It also could allow some user in developing another versions (private or public) from a previous point of the collaborative product.

- **Argumentation**: it can enable novice users to augment about their contribution in the product that are being collaboratively developed. Argumentation allow members of a group to understand the motives or reasons of a given action or contribution. It stimulates a more structured and direct communication and a negotiation process between users as well.

These criteria were summarized in a global analysis among the related system and is discussed in the next section.

2.6.2 Synthesis

A comparative analysis is summarized in table 2.1 providing some inspiration towards additional development and new directions concerning cooperative music experimentation by novices on the Web. Although there is no established classification for this comparison, as far as one knows, due to a heterogeneity of the systems, even considering them as part of the Web-based application for musical collaborations.

Table 2.1: Analyzing the Related Systems

	JamSpace	Pitch Web	EduMusical	WebDrum	Daisyphone	PSO
Sound format	MIDI					
Musical process	jam session	performance	composition	performance	performance	performance
Music metaphor	scratch tracks	geometric shapes	piano roll	grid	dots, circle	bouncing balls
Sound Exportation	-	-	.mus	-	-	-
Architecture	Client-server					
Access	restrict	free	restrict	free	free	free
Interaction	synchronous, asynchronous	synchronous	synchronous, asynchronous	synchronous	synchronous	synchronous
Communication tools	-	chat	chat	chat	shared screen (draw)	-
Persistence	-	-	implemented	-	implemented	-
Group memory	-	-	-	-	-	-
Awareness	synchronous	-	-	-	-	-
Interaction trace	-	-	-	-	-	-
Argumentation	-	-	-	-	-	-
Authorship	-	-	-	-	-	-
Requirements to run	software hardware LAN	ShockWave, QuickTime, Beatnick	Java	JSyn	Java, Beatnick	Java
Target public	musicians, novices	composers, novices	OSESP students	novices	novices	novices

The comparison is presented according to criteria associated to Web-based musical cooperative activities for novices and the fundamentals described in this section. Thus, the related systems are located in the first row and the adopted criteria are in the first column. The symbol "-" means a criterion that is not implemented by the application, or is information not available in the literature.

In the table 2.1, the first row indicates the **sound format** required by the applications. All of them use MIDI technology for sound manipulation purposes.

Most of the systems are designed for "performance" activities. This **musical process** possibly occurs in real time and is conceived at the moment of execution. Thus, important issues related to asynchronous interaction, such as persistence of the musical piece after the performance, control of user actions, and the current state of the musical piece, are not usually considered in the projects.

Many different approaches are implemented as interface metaphors to represent sound and musical information, as can be seen in the **sound representation** row. The use of common metaphors such as grid, balls, and geometric shapes, in some cases, like the "PitchWeb" and "PSO" facilitate and encourage the interaction for lay users. However, when in contact with these metaphors, skilled users (musicians) are able to produce better result depending on the implemented computer music concepts (synthesis, filters, pitch, etc.) or musical (scratch) tracks, in the case of JamSpace.

Sound exportation means the possibility of saving or exporting the current musical piece in a sound format. Most of the systems do not care about this issue, except for EduMusical, which uses its own format and is thus able to save MIDI files.

In terms of **Architecture**, the table shows that client-server is the most prevalent, adopted in all applications.

Regarding the accessibility point of view, **Access** indicates if the user has "free" or restricted access. With the exception of EduMusical, a system developed to be used by OSESP students, and JamSpace (in progress), all the systems are free for use. However, EduMusical enables free access in individual edition mode.

The criterion **interaction** aims at comparing the synchronous and asynchronous issues. All the systems implement synchronous interactions except EduMusical, which also support asynchronous interaction. This criterion has strict relationship with the **communication tool**, since synchronous systems use chat as a means of real time communication between users. Daisyphone provides a chat with graphical annotation, which is different from other systems using traditional on-line text chat. As a consequence, it can distract collaborators from their joint action and reduce their mutual engagement (BRYAN-KINNS et al. 2006).

On the other hand, asynchronous systems should provide a kind of **persistence** in order to allow users to keep track of their contributions. This is the case of EduMusical and Daisyphone, even though in the latter, regarding persistence, people became bored by the interaction affecting their engagement (BRYAN-KINNS; HEALEY; LEE 2005) .

The criterion **Requirement to run** aims at listing the necessary setup before playing the systems. In some cases, the amount of plug-ins or the use of non-standard technology may become a barrier or hard task for novices to use the system.

Most of the criteria related to collaboration as **group memory**, **awareness**, **interaction trace**, **argumentation**, and **authorship** are not considered by the

systems, except JamSpace that allows users to see what another person is playing in real-time, or the contents of a looping track.

Interaction trace corresponds of allowing users to know each other's actions inside the application by means of some logging mechanism. Argumentation is related to design rationale concept in which users associate comments with actions or decisions. Authorship, in the context of this thesis, stands for the possibility of users to know their participation (each individual contribution) in the whole musical piece.

Although all systems are conceived to allow cooperative musical interaction by novices, the comparison in the Table 2.1 shows important gaps, in particular:

- the criteria related to an effective collaboration, such as argumentation, authorship, interaction trace, awareness, group memory, and persistence, are not adequately considered or explored. Systems are not capable to allow argumentation or negotiation for users to collaborate in musical creation. Also, systems do not provide adequate awareness mechanism or trace of interaction to give adequate feedback, so the group can understand users' contributions;

- there is a need for integration of concepts from other areas in order to design useful and usable systems for novices use in musical domain, such as usability, accessibility, interaction design, awareness, and design rationale among others;

- current systems do not care about authorship of user contribution not allowing them to know their original contributions at anytime.

Considering all these gaps, the main drawback of many works in this field (networked music) seems to come from the lack of focus on the effective cooperation and the user interface conceived for novices as well. Also, there is a lack of knowledge about requirements, real needs and tasks of the novice users for cooperative musical interactions. The CODES features and the *Cooperative Musical Prototyping Process* proposed in this thesis are the efforts to fill in these gaps.

3 CODES OVERVIEW

This section aims at presenting an overview of the CODES environment, by showing the CODES design process in section 3.1, describing the assumptions in section 3.2, general requirements in section 3.3, the architecture of the system in section 3.4, and the general features in section 3.5. In section 3.6 the basic functionalities and levels of CODES are presented, the section 3.7 discusses about the need of assistance for novices in music and finally, section 3.8 synthesizes this chapter.

3.1 CODES Design Process

In general, the basic idea behind iterative design is to develop a product (wheter a car, software or a Music Prototype) incrementally, allowing the developer to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the product.

The CODES design process required attention to a number of considerations such as characteristics of the user, context, purpose, and minimal technology requirements, and the nature of its possible influence on the novice user.

A simple replication of musician oriented concepts, interfaces, symbols and features, without a careful analysis of their requirements and world views, could result in a system that would seem useless and unusable to these users. It is important that the creation of this kind of system for such groups (novices) develops in continuous partnership and dialogue with primary them, and includes a investigation into their concerns and expectations.

This process, carried out between 2005 and 2009, adopted a user-centered and incremental design approach as the basis for the CODES design. The study included iterative phases of understanding requirements (see details about general requirements in section 3.3 and novices requirements in section 4.4) and testing (see chapter 5), and was divided into two main and broad steps:

"Step 1": Investigates what CODES features novices users would like through an iterative process of questioning, testing, and prototyping using hard-broad models. Some excerpts of these models can be seen in Appendix C, section 6.4.

"Step 2": Looks more broadly at context of use, since it has involved to know, analyze, and chose the primary and suitable concepts of musical system for experts (presented in section 2.1), trying to incorporate them in a high level transparently for lay people. In addition, CODES scenarios were created containing several context of use. They can be seen in the Appendix C, section 6.4.

A needs analysis stage was followed by prototype development and testing. After identifying the needs through interviews and testing (sections 5.1.5, 5.2.5, and

figure 5.1), mid and high fidelity prototypes were developed and tested again. At each iteration, design modifications were made and new capabilities, functions, and characteristics were added, in a cyclic process.

This way, the design process adopted here enabled this research to obtain some assumptions (presented in the next section) and information directly from users on their needs and experiences of the system in question.

This kind of process is all the more important in the case of designing for novice users in music as their life experiences are likely to be markedly different from those of the musicians and developers. Involving the user throughout the design process also helps in testing and successive versions of the system's prototype by exposing the shortcomings and strengths in the design.

3.2 Assumptions

CODES - COoperative Music Prototype DESign, is a Web-based environment for cooperative musical prototyping that aims to allow novice users to experiment with music and interact with each other in order to create simple musical pieces, herein called *musical prototypes*.

Using interactive features of CODES, users can create, edit, and share musical prototypes in their group or on the Web. These shared musical pieces can be repeatedly tested, listened to, and modified by the partners, who will be cooperating on the prototype refinement.

Some assumptions that guided the development of the CODES project are:

- Music as a social activity. This research aims at allowing engaging musical experiences by novices by means of effective cooperation;
- It is not necessary any formal musical knowledge, in order to make musical experiments;
- Novices in music want to create and share their musical experiences. They only need adequate mechanisms and support;
- High level musical representation is necessary to allow high level manipulation of musical pieces by novices;
- Metaphors can be used to allow novice users to interact with musical interfaces;
- Current technology offers great contributions to social ways of music making, since it makes it possible to have virtual meetings (on the Web), also the use of virtual instruments (computers), and offers alternatives for musical representation, necessary for novices making music;
- Support for effective cooperation enables users to discuss and decide about their music and stimulates the creation of their own musical culture.
- An effective environment on the Web for shared musical experiences for novices in music is possible only if it includes characteristics for interaction, musical and cooperation activities;

These assumptions have influenced to find out the general requirements of the system, presented in the next section.

3.3 General Requirements

In the case of novices in music, there are aspects of the interface and interaction that should be specified and adapted to allow them to perform tasks of music experimentation like usability, accessibility, and awareness. In this section, the requirements of user interfaces for musical activities (including networked ones) are investigated and discussed, particularly focusing on the necessary distinction between interfaces for musical activities and interfaces for musicians.

Usually, computer music systems are designed for experienced musicians, and with rare exceptions, e.g. the networked music systems PitchWeb (DUCKWORTH 1999), Daisyphone (BRYAN-KINNS; HEALEY 2004), and PSO (BARBOSA 2005), they require previous mastering of specific skills, and knowledge of specific concepts for better usage as mentioned in section 2.6.2.

As described in chapter 4, if the intention is to design cooperation and interaction so that a musical system can be useful and usable even to non-musicians, this problem must be approached from a HCI perspective, combined with concepts from Computer Music, and also CSCW fields.

To investigate what should be a musical interface for novices, it is convenient to start by considering the context of use of traditional music software, including here its user profile (which is normally that of a musician or an amateur musician). By doing so, it is possible to understand why some of the features of interfaces for musicians are only suitable for that kind of user, and to think about how to modify those features in order to suit also the non-musician profile.

First of all, musicians know music theory. They know how to read scores, the traditional music notation with its staff and musical symbols. Moreover, they know that these symbols refer to concepts like notes, rests and tonalities - a novice may not even know what these musical concepts are all about! Even alternative notations (like tablature) contain alternative symbols for the same concepts, and the problem remains: these concepts are not part of a novice's world. Notation is a hard and non-intuitive concept for any novice to learn. At least, it must be regarded as a true possibility when designing the user interface.

In addition, musicians also have theoretical and practical knowledge about musical instruments, have access to them, and know the technical issues associated in how to play them.

As a consequence of the above, typical music software often relies on traditional music representations and on metaphors from a musician's experience manipulating musical instruments.

For example, the MIDI protocol itself which is designed to interconnect digital musical instruments and computers, is based upon "musical performance event", like keys being pressed, changes in timbre and in tonality, tempo changes, etc. Even some more recent interaction styles (such as the style adopted by IRCAM's Max/MSP (CYCLING74 2009)) are metaphors of something that musicians are used to do, requiring a experienced musician's knowledge and vocabulary. As a consequence, they are inadequate for a novice.

Another example is the cooperation aspect of some so-called cooperative systems that fail in their conception by restricting their cooperative activities to a single chat session whose the contents cannot be retrieved after the session.

Besides, usability and accessibility issues should always be considered overall, in systems intended to be online such as CODES.

Based on these considerations, the novice user profile, and the goal of providing cooperation on the Web in asynchronous musical activities, a set of general requirements are defined, as follows:

a. To develop an accessible and usable environment to promote engagement between novice users for asynchronous interactions;

b. To use common interfaces (like mouse, keyboard, computer screen, and poor audio output) to allow interactions with the musical system;

c. To be capable of running with standard computational resources, i.e., without any additional resource;

d. To avoid as much as possible the installation of additional software in order to be able to use the system;

e. To be conceived for "naive users", that is, users without any previous computer music experience;

f. To use alternative musical representation to facilitate the manipulation of sound elements;

g. To use some mechanism to keep the history of users' interactions and contributions;

Analyzing these requirements and their relation with the points discussed in section 2.6.2, a set of more specific novice-oriented requirements is presented in Chapter 4 where the cooperative musical prototyping process is detailed. The next section presents the architectural level of the system.

3.4 Architecture and Persistence

Considering that the challenge is to allow music-making to be accessible for ordinary users, the design decisions should begin at the architectural level, to facilitate the development process, allowing further to concentrate on the interface aspects.

CODES is based on the classical client-server architecture for Web applications. A first architecture, defined in a previous version of CODES, resulted in the study and development of a software architecture for a Web-based environment for cooperative musical prototyping (HARTMANN 2006). In this first version, the Webwork, a Java Web-application development framework, was used. Questions related to performance and usability have shown the need of searching for a more suitable solution, and led to the need of redesigning the initial architecture.

In the current version of CODES special attention was given to aspects related to interaction flexibility and usability. One of the main goals is to implement an adequate support for manipulation of complex musical information, cooperative activities and group awareness, in order to provide an effective interaction of the users with each other and with the environment itself.

Thus, the whole architecture is still running over some frameworks to reuse solutions to well-known problems, as can be seen in Figure 3.1. In the client-side, CODES uses the Adobe® Action Script and .swf files embedded in the standard HTML.

On the server side, CODES implements the Model-View-Controller (MVC). It is a model used to separate the logic of the application in different parts with different responsibilities. In MVC, the "Model" part (Apache with PHP) connects the Web server with MySQL database, and represents all the information (the data) of the application; and the "Controller" part manages the communication of data and the business rules used to manipulate the data to and from the model. For this, CODES makes use of Adobe® MXML (an XML-based language used to lay out user-interface components for Adobe Flex applications). This allowed the development stage to focus on the view part of this framework to deal with interface aspects.

The sound files used in CODES are small samples of MP3 files which can be quickly downloaded by the client-side and assure a standard audio quality.

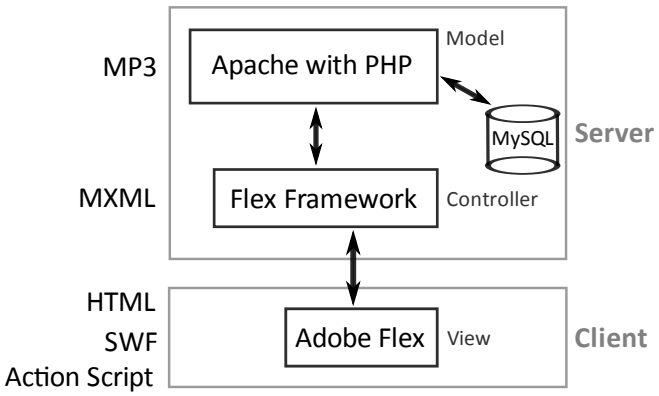


Figure 3.1: CODES current architecture

On the client-side the "View" part corresponds to elements of the user interface such as text, buttons, canvas, icons, checkbox items, and so forth; the GUI is made as simple as possible for running on a Web browser. For that, the Adobe Flex[a] script language (ADOBE 2009) was chosen to allow actions like drag-and-drop, use of sliders, scalable windows, and other facilities to manipulate the sound samples provided for this technology.

Flex is a highly productive, free open source framework for building and maintaining Web applications that deploy consistently on all major browsers, desktops, and operating systems.

Regarding Figure 3.1, in CODES the MVC model runs as follows: 1) when a user interacts with the interface (the "view" part running the .swf file in the client-side) to send comments related to some contribution, for example, the "controller" (.mxml or .as files in the server-side) manages this event coming from the user interface and accesses the "model" to update the MySQL database, based on the user action. The "view" part gets the data from the model part to update the user interface, and so on.

After identifying general and novice-oriented requirements (see sections 3.3 and 4.4, respectively) a database schema was defined. The CODES database must reflect all the necessary entities to support the cooperative musical prototyping on the Web. This schema can be materialized in any relational database system. In current CODES implementation the database system used (see the figure 3.1) is the open source MySQL. Such entities include all the stored users and all information

about them (nickname, e-mail, preferences, etc.), groups, musical prototypes, sound patterns, musical styles, versions, contributions and layers, arguments, and ratings. This way, all the predefined elements (sound patterns, for instance) for musical creation and all the other elements created by the users (drafts, new musical prototypes, versions, etc.) are stored in the database. The CODES database schema is shown in Figure 3.2.

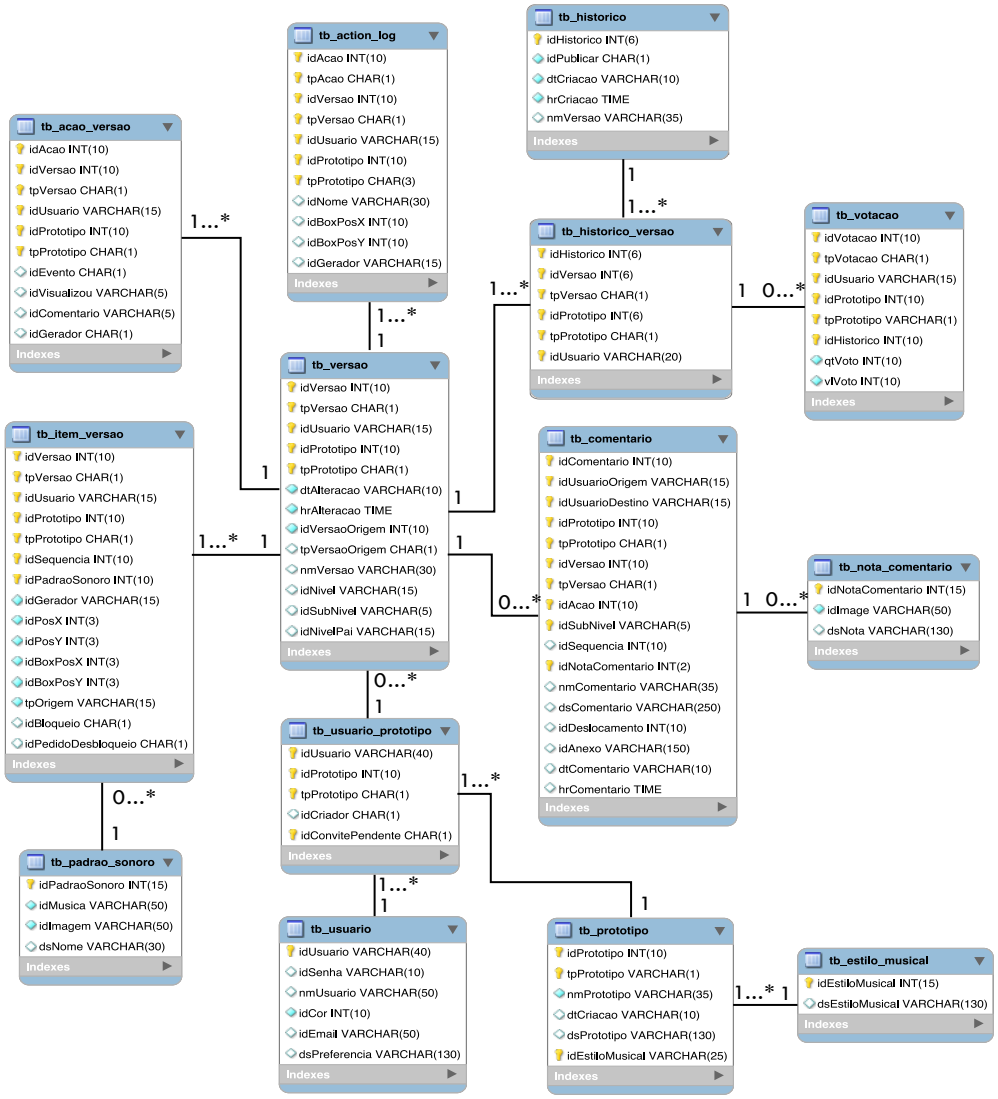


Figure 3.2: CODES database schema

Looking at the CODES database schema is possible to observe that most of required information by the system are related with the entities "tb_usuario", "tb_prototipo", "tb_versao", and "tb_historico".

The "tb_usuario" table stores the user access data, identification color (used to identify the object owner in the system), and e-mail (to receive notifications about new events). The data about the musical prototype (name, description, and musical style) are stored in the table "tb_prototipo".

From the relationship of both tables (“tb_usuario” and “tb_prototipo”) is possible to store and retrieve the list of pending invitations and of the musical prototypes in which users participate.

In the tables “tb_item_versao” and “tb_versao” are stored the data about the user contributions, such as sound patterns and musical sequences. However, the table “tb_versao” also has relationship with other tables, namely “tb_comentario”, “tb_acao_versao”, “tb_action_log”, and “tb_historico”. The table “tb_comentario” stores the information changed during the prototyping process. The cooperative actions (new comments and versions) are stored in the table “tb_acao_versao”.

The table “tb_action_log” registers all the events performed in the shared space, such as adding, moving, and excluding sound patterns. At last, the relationship of the table “tb_versao” with the table “tb_historico” results in storing the musical prototyping variations (different groups of versions) and if them are able to be rated and listened by the CODES non-members.

One of the main features that makes CODES different from the related systems presented in section 2.5, is the negotiation and contribution control mechanisms that are being held in a non-technical domain like in music. These resources enable the development of intuitive artistic activities, without the need for following rules, hierarchies, and systematization typical in traditional cooperative approach for technical domains.

3.5 General Features

Creating, editing, sharing, and publishing are the four main features of CODES. A system overview and its general features can be accessed by the hyperlink called *Tour in CODES*, available at CODES home page. (see Figure 3.3).

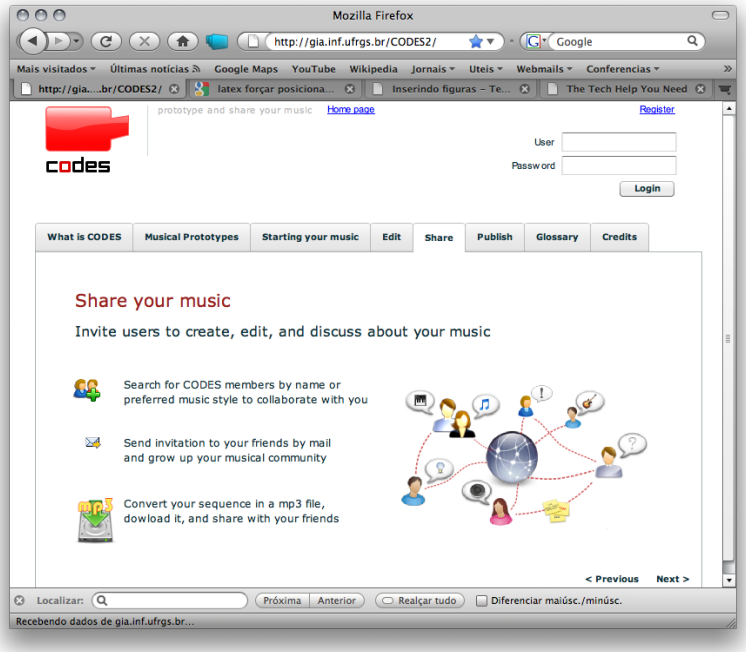


Figure 3.3: A screenshot of the “Tour in CODES” window

Users can start on a new musical prototype by choosing the name and the musical style they want. The selection of a musical style allows CODES to filter sound patterns to be offered to the user. However, since all the styles are available from the sound library , mixing sound patterns from different styles in the same musical prototype is still possible.

Edition in CODES includes actions like "drag-and-drop" sound patterns from the sound library to the editing area, "move", "organize", "delete", "expand" the duration, and "collapse" them to listen to the final result. More details can be seen in section 3.6.2.

For sharing a musical prototype, the "owner" user can invite CODES users to use a search engine or by sending explicit invitations via e-mail to non-members and asking them for cooperation. When someone accepts such an invitation, the user becomes a prototype partner and can edit it like the owner does. See more details in section 4.6.2.

At any time users can listen to the musical prototype and link arguments to their decisions, in a structure similar to that of a design rationale (described in section 2.2.3). Thus, all prototype partners can discuss and change ideas about each step of the prototype refinement, in order to understand someone else's decisions. More details in 4.7.1.

When someone considers that the resulting sounds are good, a "publication request" can be triggered and the group may discuss and deliberate about the publication of this musical prototype in the CODES home page. This activity is called *musical prototype publishing*. As an alternative to publishing their music, users may export (download) their musical prototype in an MP3 file format, and share it as they want.

3.6 CODES Levels and their Basic Functionalities

The CODES interface was designed to strike a balance between user interfaces that are so easy-to-use that they end up depleting their expressiveness, and others that are so complicated that they discourage beginners. The CODES user interface has three main levels of interaction for different user profiles: a) Public Level, b) Musical Prototype Editing Level, and c) Sound Pattern Editing Level.

Basically, the two different user profiles are CODES members (registered users) and non-members (web users). The user activities and the characteristics of the levels are detailed in the next subsections.

3.6.1 Public Level

At this level, anyone (including non-members) can access and interact with the system by exploring musical prototypes, by searching, listening, and so on. Figure 3.4 shows a screenshot of the CODES Public Level.

Regarding Figure 3.4, users can login or register, if non-members, in the login area located in region (B); explore the published musical prototypes in region (D) by listening (see example of the Hommer Song in which the stop button, by being noted as available indicates that the file is running); they can also view the prototype's information, rate the preferred ones, as well as do a search and filter, in region (C), by titles, names, rate, and musical style.

One of the goals at this level is to encourage the audience into becoming CODES

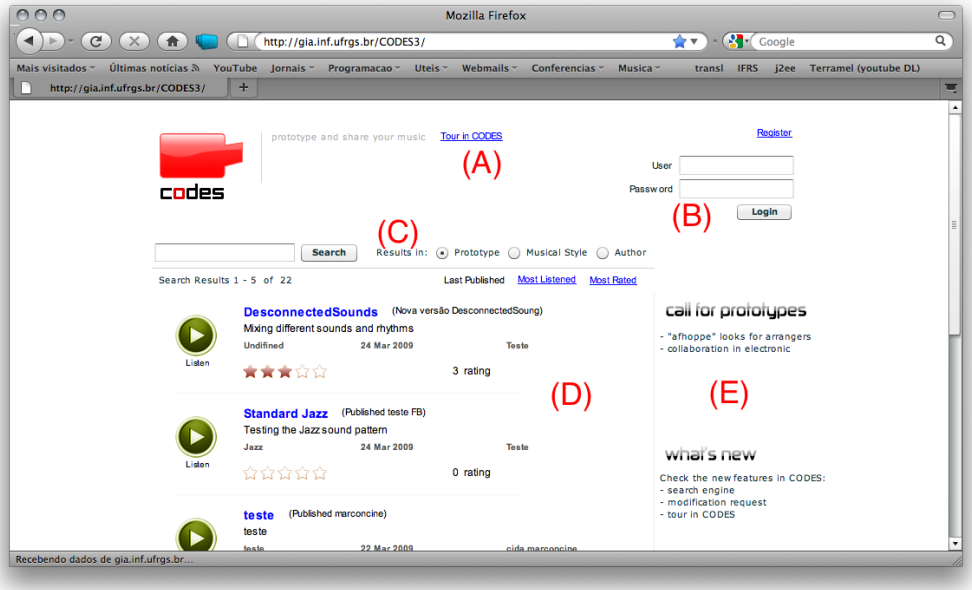


Figure 3.4: A screenshot of the CODES Public Level

members, and encourage members to publish their musical prototypes in order to engage users more and more towards the self organization of a virtual community of music.

3.6.2 Musical Prototype Editing Level

This is the most important level of the system. At this level, registered users can create and edit their MPs cooperatively. See a screenshot of the MP Editing Level in Figure 3.5.

The edition of a MP in fact is a simple task. Looking at the Figure 3.5, it is possible to identify some regions of the interface at this level. For instance, the sound patterns are dragged from the sound library - region (D), and dropped into the MP editing area - region (B).

At any time, the user can create a new MP- region (A), or play an existing one - see the execution control buttons in region (C) of Figure 3.5. The sound patterns displayed in the editing area are played from left to right. By pressing the "Play" button, a vertical timeline runs left to right, giving feedback to the user as to what are the sound patterns which are being played. It runs until the last sound pattern positioned in the editing area and then goes back to the beginning, in a loop mode, until the user presses it again to stop the execution. This is a very important feature of CODES since it allows users to edit and try their sound sequences while the system is still running.

The basic actions at this level is to add or remove sound patterns in the editing area, as well as change their sequence, size, combination, and position. Actually, these actions are included in the *Music Prototyping Process* presented in section 4.5.

Each author's contribution in the shared workspace is identified by color: the edges of icons of sound patterns are colorful (with the same color chosen by the user at the registration).

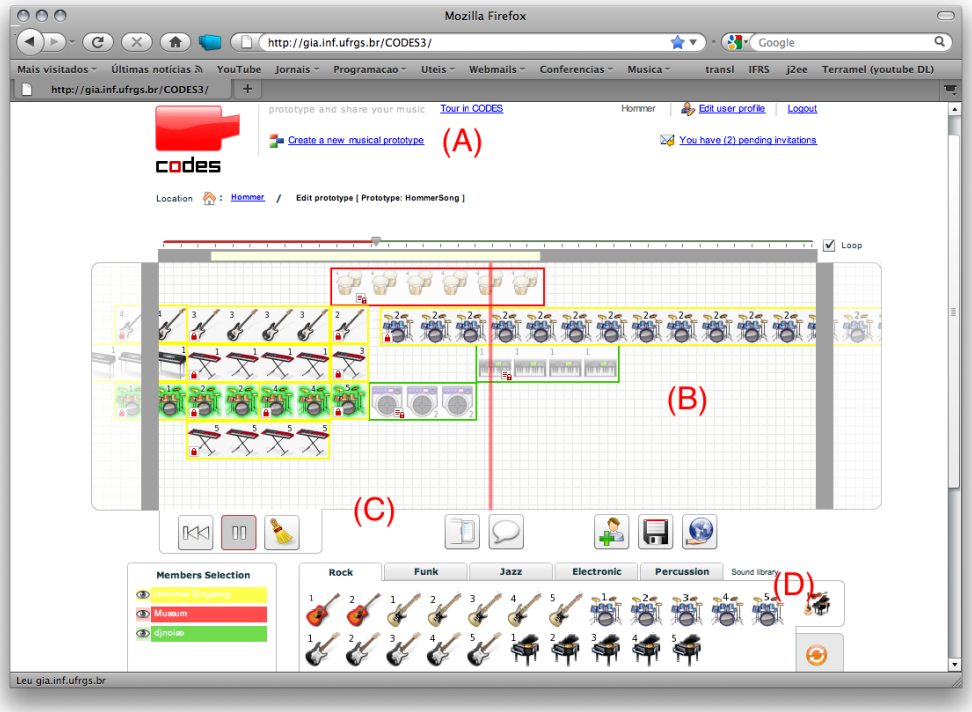


Figure 3.5: Musical Prototype Editing Level

In the members area, a user may show or hide other users' contributions (in fact, other users' layers) by clicking over the colored user Id. These actions are part of group interaction and authorship control mechanisms of CODES, described in sections 4.6 and 4.8, respectively.

It is possible to listen to each layer separately, compare, combine contributions, ask to edit others' contributions, and of course save the result.

Each new action performed in the system is stored in a database log. CODES uses these data to give a visual feedback to other users. Detailed explanations about awareness mechanisms of CODES are presented in section 4.7.

3.6.3 Sound Pattern Editing Level

This is the lowest level of CODES for member users and aims at providing total flexibility by enabling the option of editing the elements (in fact, notes) of the sound patterns, by means of a "piano roll" editor (ROADS 1996) , as shown in Figure 3.6.

When a sound pattern is opened for edition in this level, the editor represents its notes as horizontal bars. The vertical position in the grid means the pitch of the sound, and the length of the bar, its duration. So, the highest the bar, the higher the pitch. The longest the bar, the longest the duration of the sound as well.

Thus, using the mouse pointer to drag the bars, users can easily change this values and check the resulting sound. There are also interesting possibilities other such as change the timbre of the sound and editing facilities like select, copy, and paste blocks of notes. Novice users have the option of "try" and "error" to obtain the desired sound in a virtual piano keyboard, representing a different abstraction

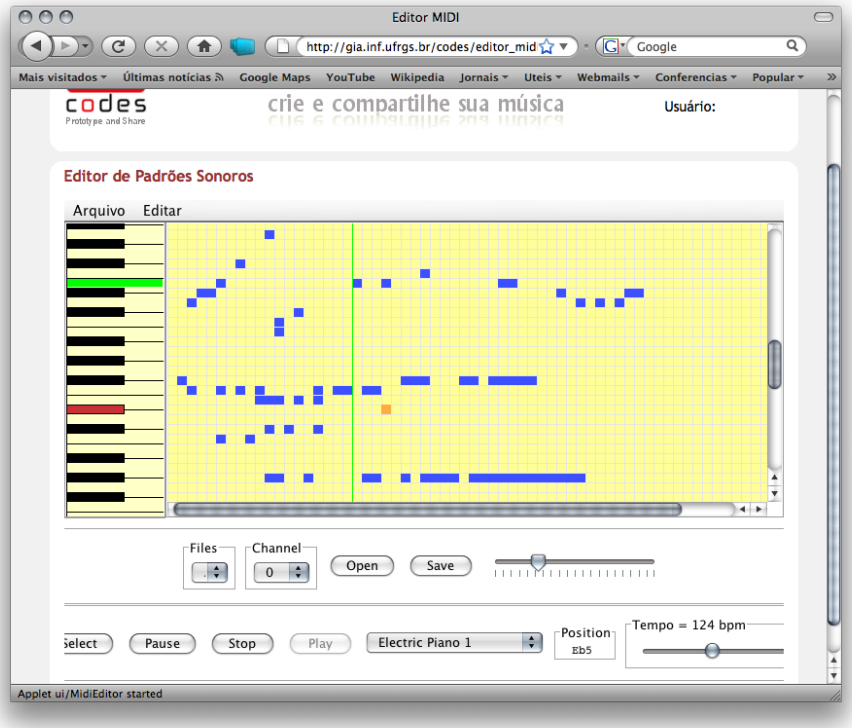


Figure 3.6: Screenshot of the Sound Pattern Editing Level

of the sound (staff, tablature, etc.). This kind of representation allows novice users in someway to manipulate the music notation in a high level of interaction, as well as editing notes as they go.

It is important to mention that the sound pattern editing level, even it has been developed in a previous version of CODES, it is not yet integrated in the current version and also was not tested in CODES experiments in Chapter 5.

3.7 Assistance in CODES: The CODIVA Agent

This part of the research reflects an attempt at integration of CODES with an Assisting Conversational Agent dedicated to the function of assistance to ordinary people involved in the collective musical experimentation provided by the system.

This research was held at LIMSI[1] (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur), Université Paris-11, during the author's doctorate internship in Paris.

The research group Architectures and Models for Interaction (AMI) has a multidisciplinary team of computer and information scientists, sociologists, cognitive scientists, ergonomists and linguists working with computer-mediated interaction (specially with Conversational Agents), under coordination of the Professor Jean-Paul Sansonnet.

When a sophisticated RIA like CODES is considered, a more drastic approach to assistance is required, featuring high level help techniques. Moreover, beyond

[1] http://www.limsi.fr/index.fr.html

the basic GUI-related problems, making people learn and progress in the task of producing music samples is also within the scope of interest of this thesis, so that they can evolve from passive music listeners to creators of music content.

This implies specific requirements for this kind of help systems and makes them play a primary role in Web-based distributed collective practices. Therefore, this thesis looks for a suitable solution on the function of assistance in order to provide CODES with a powerful help system, specially designed to work with ordinary people.

The research focused on Assisting Conversational Agents (ACA) (LERAY; SANSONNET 2007), a kind of Embodied Conversational Agent (ECA) (CARROLL; ROSSON 1987) dedicated to assist novice users to interact with general public applications.

To this purpose, the DIVA toolkit (SANSONNET 2009) was integrated into CODES in order to support the use of assisting virtual agents on the Internet.

From that emerged the CODIVA (CODES + DIVA) whose objective is to study the notion of music-assisting agents in the context of the Web-based distributed collective practices, mainly involving ordinary people who are playing, socializing and learning over the Internet by exchanging and producing musical content.

This integration was carried out in three main phases:

1. **Web architecture**: Technical definition of integration approach of the DIVA agent into the CODES architecture;

2. **Assisting agent**: The second phase is the synthesis of a specific assisting agent, aware of the operations and the tasks available in the CODES Graphical User Interface. Using the CODIVA framework, a first experiment with human subjects has permitted to exhibit the main features required for the design of the assisting agent. More details about this experiment can be seen further in section 5.3;

3. **Musical tutor**: The next phase will be to afford the CODIVA framework as part of the ongoing e-learning project on Pedagogical Rational and Affective Intelligent Agents (PRAIA[2]). This issue is not discussed in this thesis but it is mentioned here to give an outlook of the span of the research that CODIVA is involved in.

3.7.1 Architecture of the DIVA NLP-chain

In order to study the Function of Assistance in the context of CODES, an experimental toolkit called DIVA was developed, which is freely available for research and education purposes (SANSONNET 2009). DIVA stands for "DOM-Integrated Virtual Agents", which emphasizes its full Web 2.0 approach to assisting tools: the toolkit is completely written in JavaScript for the support of a) the virtual characters that personify the assisting agent, b) the NLP-chain that analyzes and resolves the users' questions, and c) the AJAX link to the server for access to the resources and client information storage.

The general architecture of a DIVA assisting agent is given in Figure 3.7.

[2]An international project of cooperation between the research groups PPGC/UFRGS and PIPCA/UNISINOS in Brazil and LIG and LIMSI in France.
Available at http://gia.inf.ufrgs.br/praia/

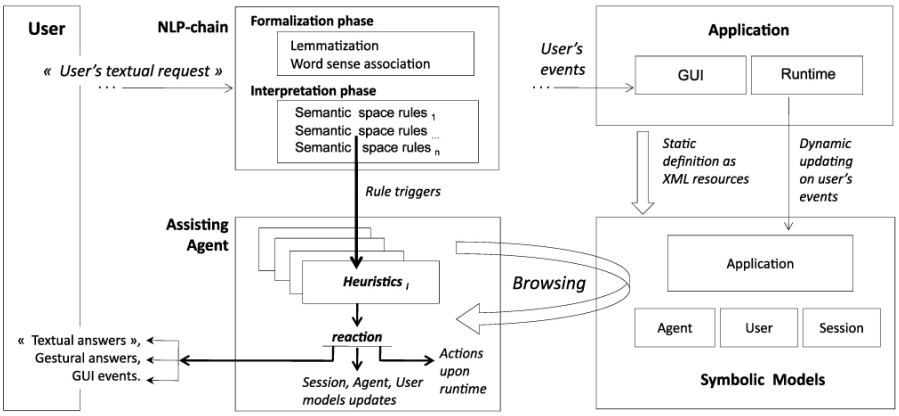


Figure 3.7: General architecture of a DIVA Assisting Agent

The NLP-chain of the DIVA toolkit is based on a typical chatbot approach but with a sophisticated structure as shown in Figure 3.7 (middle-top):

1. The formalization phase: based on two sets of filtering rules applied sequentially:

   - Syntactical level: first a typical chunking phase is applied, then words' inflections are transformed into their corresponding lemmas (root words);
   - Word-sense association level: lemmas are then transformed into semantic "synsets" as in WordNet (FELLBAUM 1998).

   At the end of the formalization phase, the request is transformed into an intermediate formal form, called the Formal Request Form (FRF). In the FRF language, a request is expressed by a sequence of abstract keywords, each of them being associated to a semantic concept defined by a textual gloss.

2. The interpretation phase: based on a set of rules of the form pattern $\rightarrow$ reaction, where the pattern is expressed in FRF, and the reaction is a procedural heuristic defining the behavior of the agent in response to the user's request. To build a reaction, the triggered heuristic uses two kinds of information: a) a representation of the current dialogical session, and b) a symbolic model of the application describing its specific features. The set of interpreted rules is organized into so-called semantic spaces dedicated to specific domains.

The Figure 3.8 shows an illustration of the CODIVA indicating the login area after the user click with the mouse in this region.

An experimental protocol related to CODIVA is then presented and discussed in section 5.3.

3.8 Synthesis

This section has discussed some requirements of novice-oriented user systems for musical activities and has also presented the Web architecture suitable for supporting collective creation of content and group interaction.

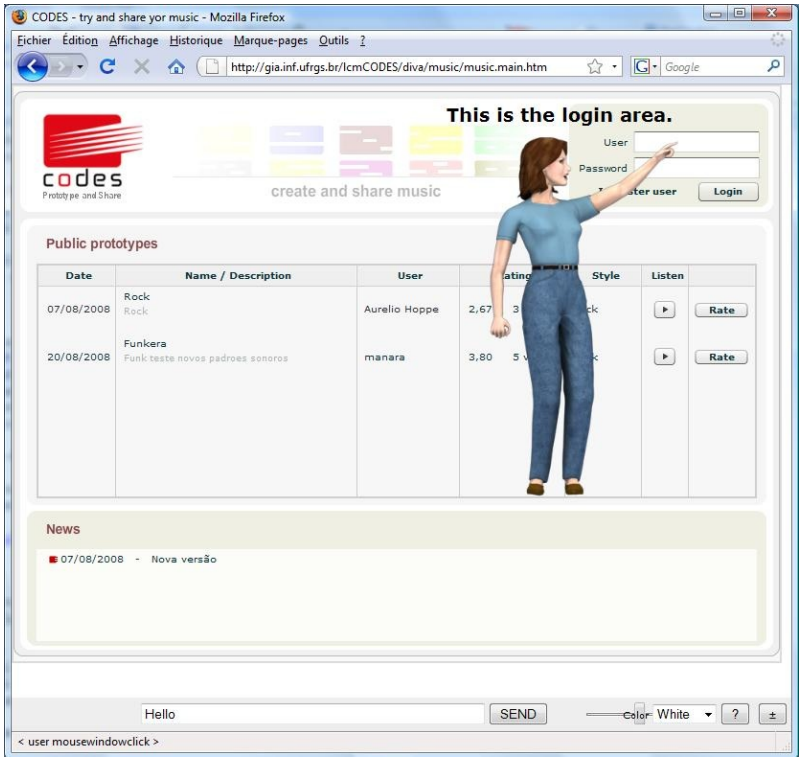


Figure 3.8: CODIVA - integrating CODES and DIVA

The overall design philosophy of CODES is to begin with a specific technological platform (Web) and application area (musical prototypes), and then leverage their affordances to find possible interactions that address the requirements of the scenario of cooperative musical prototyping.

The CODES approach for cooperation among users, in order to create collective MPs and to allow argumentation and discussion, enables each user to understand the principles and the rules involved in the complex process of music creation and experimentation.

Through computer technology, CODES provides an effective way for breaking some barriers for novices who wish to engage themselves in musical prototyping and experimentation. The point here is the mere possibility of "creating it" and not of the quality of the finished work, as already mentioned in section 1.3.

In addition, CODES has also investigated the use of the Function of Assistance (particularly ACA), a widely known solution used by students in learning situations. However, this function applied in the context of helping lay users in musical tasks is not usual at all.

For CODES project, the challenge has been to provide support for musical activities without such an orientation on a musician's reality, because novices clearly do not act and think like musicians do. Moreover, the metaphors and concepts usually adopted in computer-based environments for representing elements of musical activities (notes, melody, harmony, tempo, rhythm, timbre, etc.) in general are not well understood by novices. It can be seen as a good starting point to the definition of a specific novice-oriented process to create cooperative music, as presented in the next section.

4 COOPERATIVE MUSICAL PROTOTYPING USING CODES

This chapter describes how CODES provides support to the collective music creation process by ordinary users over the Web, defined in this thesis as *Cooperative Musical Prototyping*. An MP is an artistic product which should be created as a consequence of a prototyping and cooperative process.

One of the main issues of CODES is to provide the support for the design activity in a non-technical domain like music. Since such design is novice-oriented, it needs to provide a very specific kind of support, typically based on the *prototyping* and *cooperation* activities.

These are two very important principles (confirmed by findings obtained during CODES development and usage - see chapter 5) to be considered when providing such support as to novice-oriented music creation activities (MILETTO et al. 2009). They are: a) *Music creation by novices should be prototypical*; and b) *Music creation by novices should be cooperative.*

These principles are detailed in sections 4.1 and 4.2, and associated with the considerations about the design in non-technical domains, as presented in section 4.3, allowed the definition of novice-oriented requirements, described in section 4.4. The section 4.5 details the activities involved throughout the cooperative musical prototyping process and, further, section 4.7 presents the awareness mechanisms required for users to grasp the context of their overall activities. The group interaction mechanisms comprised in this process are presented in section 4.6. Cooperative work in music entails the need for preserving users contributions. Section 4.8 discusses the approach adopted here and, at last, section 4.9 summarizes the chapter.

4.1 The Need for Prototyping

A prototypical music creation process means that novices can draft simple musical pieces. It is called here as *Musical Prototypes* which can be tested, modified, and repeatedly listened to, in a cyclical refinement of an initial musical sketch until a final stage is reached. This process clearly resembles the prototyping cycles adopted in industry and in incremental software development. Since music creation is in fact a (music) design activity, it seems natural and straightforward to adopt a prototypical process usually adopted in design activities. In the music literature, "draft" is the term commonly applied to such kinds of creative work, but here the emphasis is on the cyclical prototyping process and not on the product itself; consequently, "prototype" and "draft" correspond to the same idea. Figure 4.1 illustrates

this comparison.

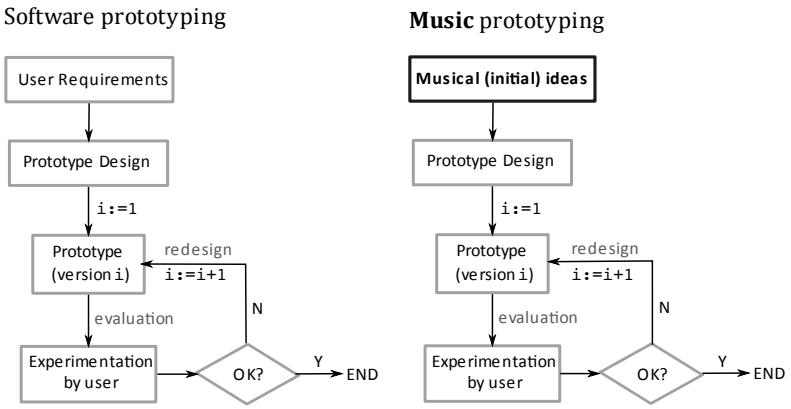


Figure 4.1: Comparing software and music prototyping cycles

Both are cyclical and include iterative experimentation by the user through n-versions until the final stage. However, there is an important difference in the final result. In music prototyping this result probably cannot be defined previously, due to its creative nature. Moreover, new possibilities can emerge from the iterations. More details about it are discussed in the session 4.3.

4.2 The Need for Cooperating

In a cooperative music creation process, the refinement of an initial musical idea is a consequence of a collaboration of the author(s) of the initial musical idea and thus of their partners; all members of a group (in fact, a social network built by explicit invitation) that are cooperating until a final consensual stage of MP be reached. Providing opportunistic negotiation through the MPR (see section 4.7.1), CODES encourages novices to be creative and try solutions, combinations, and experimentations that cross the boundaries initially imposed by the traditional musical notation.

This process is clearly a particular kind of Human Centered Collaborative Design where the result of the design is an MP.

In the viewpoint of this thesis there is a difference of the nature of the cooperative process for design in technical and non-technical domains (e.g. music) The next section explains this viewpoint with special attention.

4.3 Understanding Design in non-technical Domains

Cooperative approaches for non-technical product design, like MPs, require a very specific kind of support for collaborative activities.

In fact, the conventional cooperative approaches with fixed goals and roles, not allowing unsystematic and opportunistic negotiation, are not adequate for the dynamic, creative, and collaborative nature typically associated to collaboration in arts.

Table 4.1 summarizes the main differences found by comparing the cooperative support defined in the CODES development with some characteristics generally

found in cooperative environments for technical product design, like in manufacturing, construction, etc.

Table 4.1: Cooperative activities general framework

	Technical Design	**non-technical Design**
Main Goal	Based on a model	Creative (no models)
Group Topology	Typically hierarchical (leader)	Typically non-hierarchical
Control	Coordination	Argumentation
Planning / Decisions	Rigid and systematic	Unsystematic (negotiation)
Roles / Tasks	Fixed	Non fixed
Example	Collaboration in manufacturing	Art collaboration (CMP)

One important aspect to be considered in musical domains, specially in a group of novices, is the, *a priori*, unknown final result.

For technical products, there is a need of specifying a product model in order to standardize the process and predict the final result. On the other hand, for non-technical products, the emphasis is on the subjective aspects of the act of creation rather than on following a specific model for creation.

Considering that the final result is not unknown and there is no established models for a collective musical piece, the process is rather guided by the creation or **creativity** itself, instead of a previous design.

Besides, this process emerges from the cyclic interactions of the group, based on contributions from/to each other, and the control of the process is done by negotiation between members, without the need for a role of an explicit controller.

If all decisions are supposed to be consensual by negotiation, and not imposed by the authority of a leader, it is not necessary to make a distinct and explicit representation of the leader because usually, in a hierarchical group, the leader's opinions and actions may inhibit other users' participation. Indeed, interactions may evolve as time passes, and the more "skilled" users can be recognized and naturally respected by the group while suggesting and justifying their contributions. This allows total flexibility without the need of prior role definition, task allocation or responsibility assignment for members.

Because the cooperative music prototyping process can justifiably be seen as a political process determined by conflicts and cooperation, the joint development of ideas by means of both a multi-perspective approach and negotiation support is particularly important.

The multiple actors, who are all cooperating in the refinement of the music prototype, have different perspectives on the creative process and its results (the music prototype), each one with different backgrounds and opinions due to the context they come from.

Therefore, it is essential to support mutual understanding and to resolve conflicts during the cooperative music prototyping. A negotiation between these different viewpoints and goals must be explicitly supported and maintained over time, that's why the decision-making process is cooperative and distributed.

Actual cooperative activities are very difficult to automate and to control because they involve the complexities and the dynamic nature of human group work. This thesis is an attempt to support them. In fact, this support for cooperative music prototyping is a particular kind of Human Centered Collaborative Design.

The basic idea of CMP process is that members cooperate not only by means of explicit conversation and explicit actions on a shared objects space, but also by interpreting the messages and actions of other actors in accordance with the model of their thinking and acting, which has been built up on the course of their interaction.

A shared objects space involves prototype-oriented information, which comprises all information about music prototypes, including their composition (combination of sound patterns, versions formed by layers) and social-oriented information (including interactions between actors during the process).

Sound pattern are music samples available to users, represented by an iconic format. Manipulation of prototype-oriented information is goal-motivated. Typically, it includes prototype's elements manipulation, such as use, modification, combination, replacement, and experimentation (audio listening) of sound patterns.

Social-oriented objects are wholly related to conversation, like messages and comments. One significant consequence of recognizing social-oriented objects as relevant information is that, instead of considering modifications as only explicit transformations on an MP, their changes are also considered. That is, CODES interprets modifications on a shared objects space as meaning changes in both the MP and the social context. Thus, a sequence of messages may, at the same time, not change the MP and significantly alter some actor's argument, opinion or decision.

Unlike a conversation, where messages are categorized as to their purpose within the conversation, for action, clarification, orientation, and so forth, conversation in CODES is simply comprised of all recorded messages sent and received to/from the CMP actors, indexed to other relevant model components. Then, recording the actors' messages is extremely useful in order to capture, in an implicit way, the background knowledge, concepts, definitions, and opinions surrounding their viewpoints.

Notice that the actors cooperate via the shared objects space, that is, either indirectly by means of music prototypes that they manipulate and modify, or directly by means of conversation. Thus, the set of actions an actor may perform has been broadened to include direct interactions with other actors, in addition to traditional actions of prototype manipulation. In fact, during group activities people do not strictly act by goal-based product modification. Unless the actors in these groups are, to some extent, multidisciplinary themselves, the communication between group members plays a crucial role to support cooperative and multidisciplinary activities.

It is important to note the pedagogical potential of cooperative and creative systems like CODES that may be explained by Vygotsky's theories (VYGOTSKY 1980). This author suggests that social interactions play a fundamental role in shaping internal cognitive structures. According to Vygotsky, cognitive development derives from the person's engagement in cooperative problem solving. In these situations, the learner is forced to examine his thinking when challenged by others, and in turn to keep an eye out for possible mistakes made by his collaborators.

Section 4.6 describes how this viewpoint is put in practice by means of customized group interaction mechanisms designed for CODES.

4.4 Defining Requirements for Novices in Music

This section presents more detailed requirements inside the CMP process. The starting point is to know the obstacles that make it complicated for novices to participate in music creation. In short, these obstacles are related to the concepts presented in section 2.1, and are associated to:

a. **How to play music**? Novices need to own a musical instrument and know how to play it;

b. **How to represent music**? Novices need to represent the result of a creative process in order to repeat it later and to communicate it for anyone else;

c. **How to provide support for cooperative music creation**? Novices need group interaction mechanisms in order to not only menage group formation, but also to perceive and understand actions of partners.

Considering these questions, expanding the general requirements presented in section 3.3, and taking into account the nature of non-technical activities in non-technical domain like music, the novice-oriented requirements regarding the obstacles mentioned before are now defined as follows:

- Requirements about playing music:
  - Use conventional interaction mechanisms. Prefer not to demand sophisticated interaction devices (like complex controllers, gesture interfaces, VR, etc.), but everyday technologies (mouse, keyboard, and usual audio features available on most commercial PCs) intead.
  - Avoid conflict with musical tasks (which involve sound), by preventing sound feedback (apart from the sound being created, of course).
  - Don't forget other common usability requirements, which become even more important when focusing on non-expert users: easiness of learning, interaction flexibility, interaction robustness, and constant feedback.
  - Allow users to organize their music. In a sound library in which a large amount of samples are available, users should have the option of saving and classifying their preferences.
- Requirements about representing music:
  - Do not rely solely on traditional music notation, nor demand from users the knowledge of music theories and concepts for them to work with music. CODES has mechanisms implemented to represent sound patterns as icons.
  - Use musical metaphors from real life, known by anyone, and not metaphors from a musician's reality. Such a metaphor needs obviously to include everyday concepts and vocabulary, avoiding technical or specific terms from a musician's world.
  - Offer alternatives of music representation/encoding formats, making it easy for users to export/import their music between different systems.

  - Allow, as much as possible, for users to choose how to represent or name their own musical material. Considering the cultural deviations, the signs and metaphors can result in opposite meanings.

- Requirements about making music cooperatively:

  - Make the system multi-platform if possible, minimizing requirements of use and thus increasing user access (this is an architecture/implementation requirement, but it has an effect on system usability).
  - For a cooperative system like CODES, a very important interface characteristic should be the users' capability to perceive and analyze group members' actions on the object they are working, and to know the reasons behind each one of these actions. These are aspects related, respectively, to awareness and rationale mechanisms, which then must be provided in the interface.
  - Offer alternatives for users to search members or partners by using some criteria (names, music style, preferences, etc.) to encourage the learning process by example.
  - Offer the possibility of inviting other users by e-mail in order to share and help in the creative process. If possible, link the ongoing project in the body of the mail message (url) to allow the new users to get directly inside it to perform the cooperative edition.

This, of course, is a non-exhaustive list of requirements. Some of them seem to be obvious, but others are not so straightforward. Still, this research considers these requirements as very important ones so that the next section discusses about aspects of CMP process influenced by them.

4.5 The Cooperative Musical Prototyping Process

The *Cooperative Musical Prototyping* (CMP) is a process in which novice-users interact with each other via the Web-based environment CODES in order to create collective musical pieces, deliberately called *musical prototypes*. See Figure 4.2 for a graphical illustration of the process.

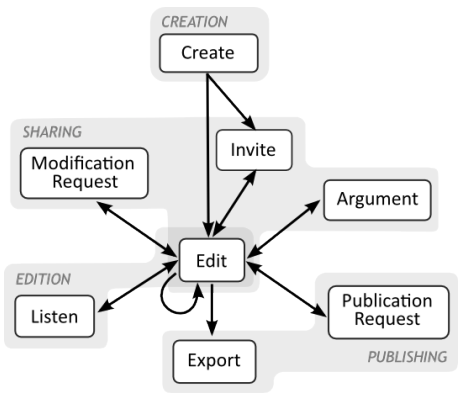


Figure 4.2: Cooperative musical prototyping process with CODES

CMP is a simple cyclical process including the following activities: a) *musical prototype creation*, b) *musical prototyping edition*, c) *musical prototyping sharing*, and d) *musical prototyping publishing*.

These activities are described as follows:

a. **Musical prototyping creation**.
The creation of musical prototypes is a simple task. It means to start on a new musical piece in which a user can name and add textual characteristics in order to be identified by the group and by Web users, if published.

Figure 4.3 shows the window for creating a new musical prototyping.

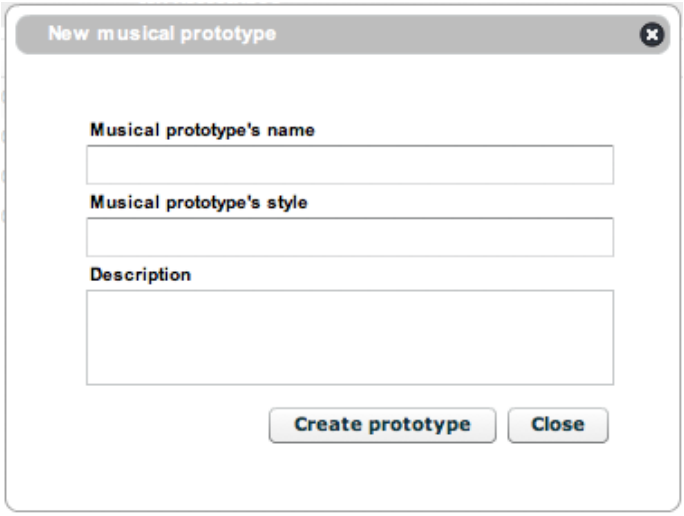


Figure 4.3: CODES window for creating a new musical prototype

The user writes the MP's name, chooses a musical style (not mandatory), and inserts a comment (not mandatory either). These information will be available in the CODES home page for users to know more about the published musical prototypes. Then, the MP may be edited, i.e., elements can be inserted in the MP to compose sequences and combinations of sounds for which any musical theory is required.

b. **Musical prototyping edition**.
The following descriptions are based on the Musical Prototyping Editing level, presented in Figure 4.4.

This figure is divided in regions indicated by letters to exemplify the descriptions.

CODES offers a high level of music representation (m) and user interface features to allow an easy direct manipulation (drag-and-drop (l)) of icons representing sound patterns for music creation.

Considering Figure 4.4, the edition of a musical prototype basically involves the sound pattern manipulation.

Users can listen to the sound patterns by clicking over them at the sound library (m) before they drag-and-drop (i) into the editing area. These sound patterns are MP3 files of four-seconds of size available at the sound library.

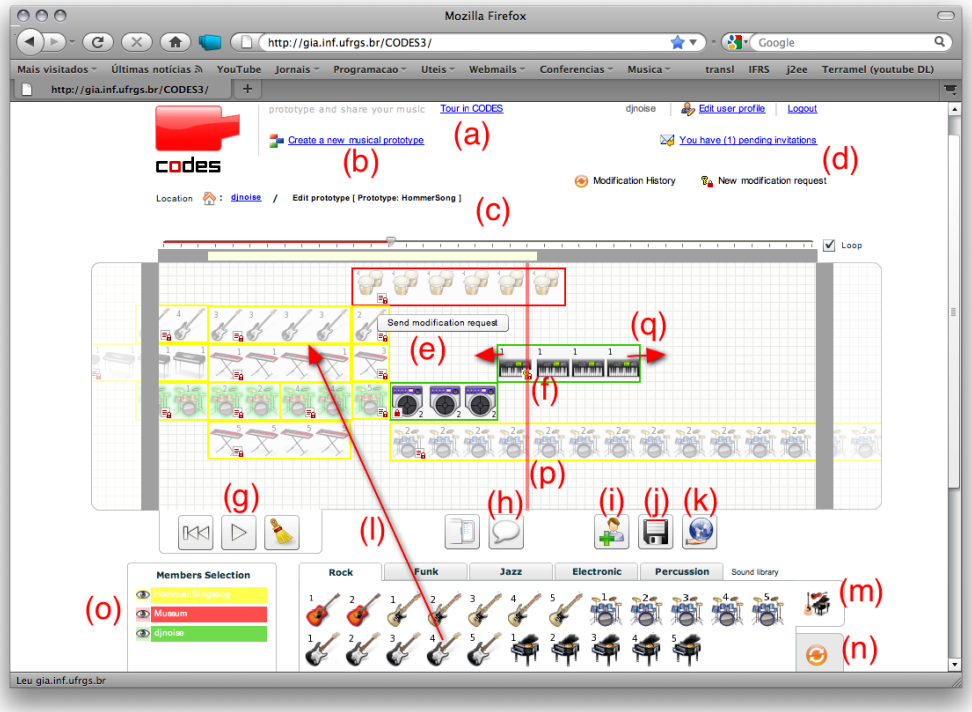


Figure 4.4: The music prototyping main functionalities

The manipulation includes activities such as changing the sound pattern position, and the actions of expand and collapse (q), delete (g), block and unblock (f) them. After the editing, users can save the work (j) for further use.

At any time, users can listen to the whole musical prototype (g) and write arguments (h), that may be linked to all decisions/activities made on it, in a structure similar to a design rationale structure.

c. **Musical prototyping sharing**.
For prototype sharing, the prototype **creator** (called the *prototype owner*) can send **invitations** to CODES members, and also to non-members, asking them for cooperation. When the invited users accept the invitation, they become a prototype partners, with all the editing functionalities enabled to them. Thus, all prototype partners may discuss and change ideas about each step of the prototype refinement, as a good way to understand someone else's decisions. In fact, a prototype's arguments (h) and modifications whose log is identified in (n) are equally considered as typical contributions in a cooperative musical prototype.

Others' contributions are disabled by default for edition in the CODES editing area. In this case, users can send a *modification request* (e) and wait for the approval of other partners to change their contributions.

d. **Musical prototyping publishing**.

When someone wants to publish the MP, a *publication request* can be sent to the group and then the cooperative musical prototype can be published in the CODES

home page. This activity is called *prototype publishing* (k). As an alternative to share their music, users may export (download) their musical experiments performed in CODES as an MP3 file, sharing it as they want.

These four main activities include the participation of partners and may be done in the context of a group for which some special mechanisms were developed. They are described in the next section.

4.6 Group Interaction Mechanisms

CODES offers different interaction possibilities at different levels. In brief, they include exploratory tasks by navigation in the CODES home page, cooperative edition for a group of registered users, and mechanisms for music prototyping rationale, publication request, and modification request. These are detailed below.

4.6.1 Interacting in the CODES home page

CODES home page was conceived to allow interaction between members and the audience. Users have available the possibility of "exploring" the published musical prototypes, "searching" them by prototype name, user, and musical style. An excerpt of the CODES home page is in Figure 4.5.

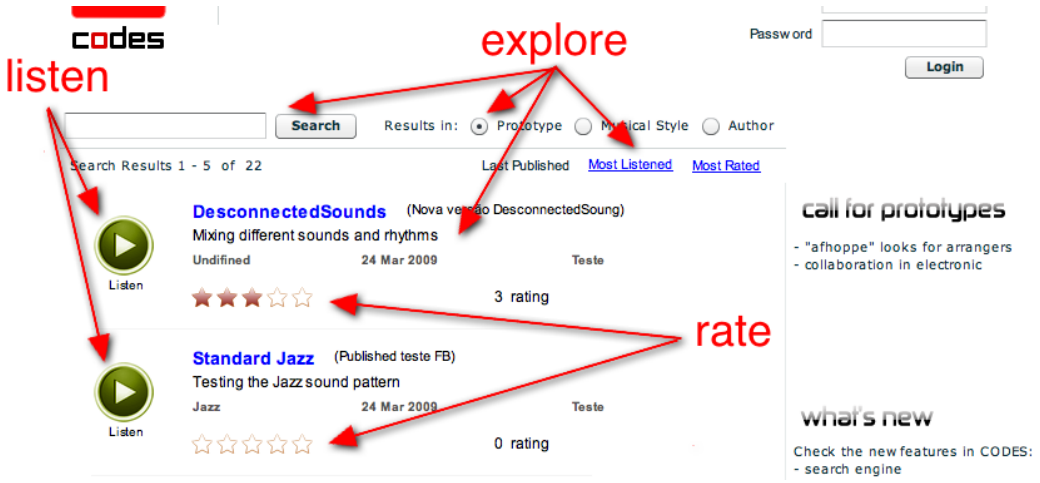


Figure 4.5: Excerpt of the CODES Public Level screen

It is possible to "listen" to MP's, classify the list of the published MP's by "most listened", "most rated", "most commented", and "last published" (default).

Rating an MP is an interesting option for CODES members to get to know about the impact of their musical creation on the audience. With the mouse cursor over the stars, users may rate the MP, choosing a value between one to five with the following labels (in order): "poor", "nothing special", "worth", "pretty cool", and "awesome". This simple example of interaction can be considered as a way to give a feedback to the author or group regarding the audience's approval. More, the ranking is also a criterion "most rated" available to sort the existing MPs.

4.6.2 Cooperative Edition

In CODES, a cooperative musical prototype is initiated by the owner who creates a new prototype, elaborates an initial contribution, and asks for the collaboration of other "partners" by sending explicit invitations. An example of the inviting window with these possibilities is in Figure 4.6 These invitations can be sent to CODES members (by searching CODES users by name or preferred musical style - Figure 4.6-1) or to non-members via e-mail (Figure 4.6-2), asking them for cooperative edition.

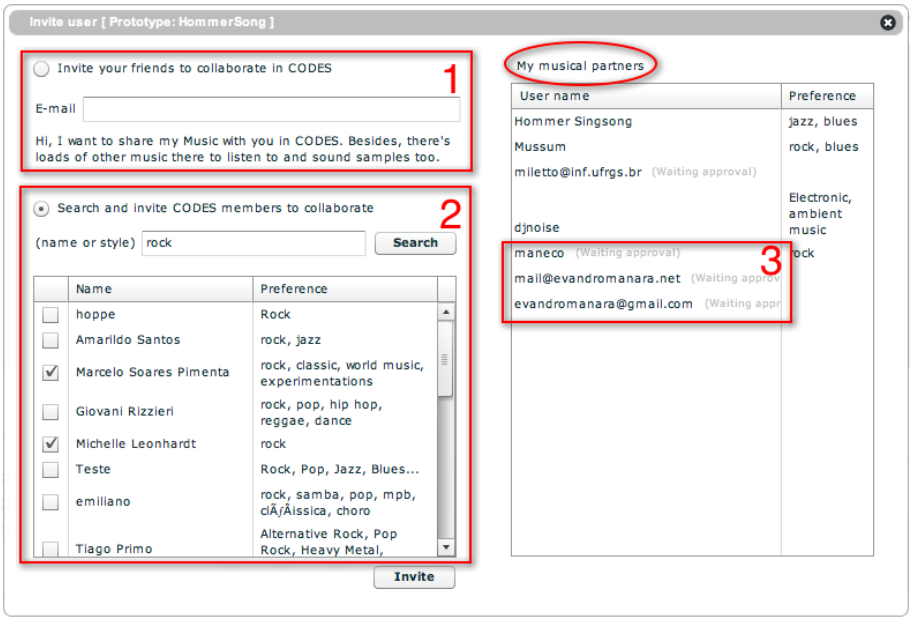


Figure 4.6: Inviting user for sharing the current MP

In the case of invitations to non-members, a hyperlink is sent in the e-mail allowing them to fill out a registration form and to get directly into the shared musical prototype to be edited. CODES informs to users the status of the invitation (Figure 4.6-3). Partners who accept the invitation can participate in the collaborative musical edition and in the prototype's refinement.

Figure 4.7 shows an example of three users, namely "Hommer Singsong", "Mussum", and "djnoise" cooperating in the same MP.

It is possible to see in Figure 4.7 the user "djnoise" logged in and his active contributions (sound patterns), while others are blurred. The members list is displayed in the members area at the bottom-left corner. Each author's contribution is identified by color in the shared workspace: for example, the edges of sound patterns icons are colorful (the color is chosen by the user at registration, in their first time use). In the members area, a user may show or hide other users' contributions (in fact, other users' layers: see section 4.8) by clicking over the user id. Thus, it is possible to listen to the contribution of each user (each layer separately), compare and combine contributions, and of course save the result. See an example in Figure 4.8 where the contributions of Hommer Singsong are unselected.

If the "djnoise" user wants to suggest or change others' sound patterns (which are blocked for him by default), he has an option of sending a "modification request" to an other user and to start a negotiation process. By clicking in the red small

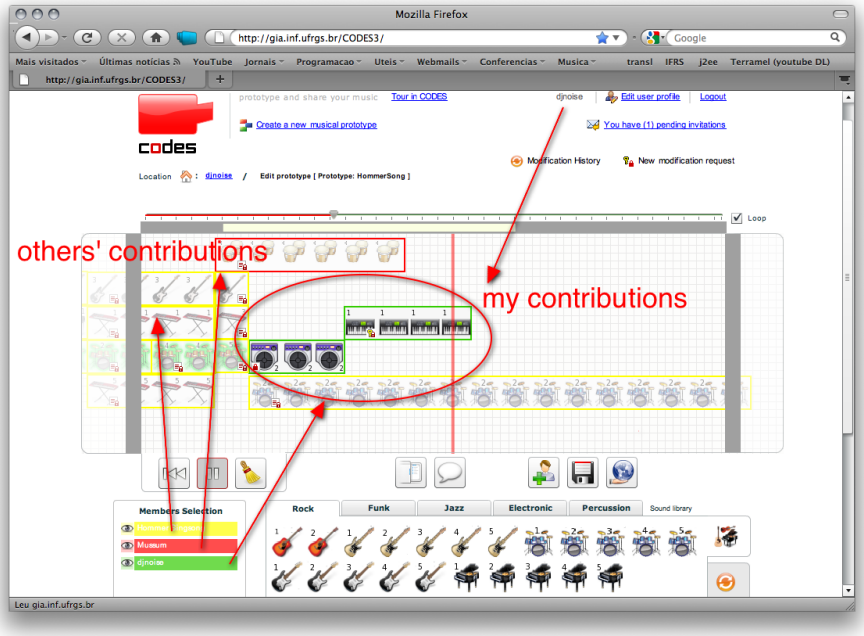


Figure 4.7: Three users cooperating in a shared MP

padlock of a given sound pattern, he will have the possibility of changing that sound pattern if his or her owner accepts the request.

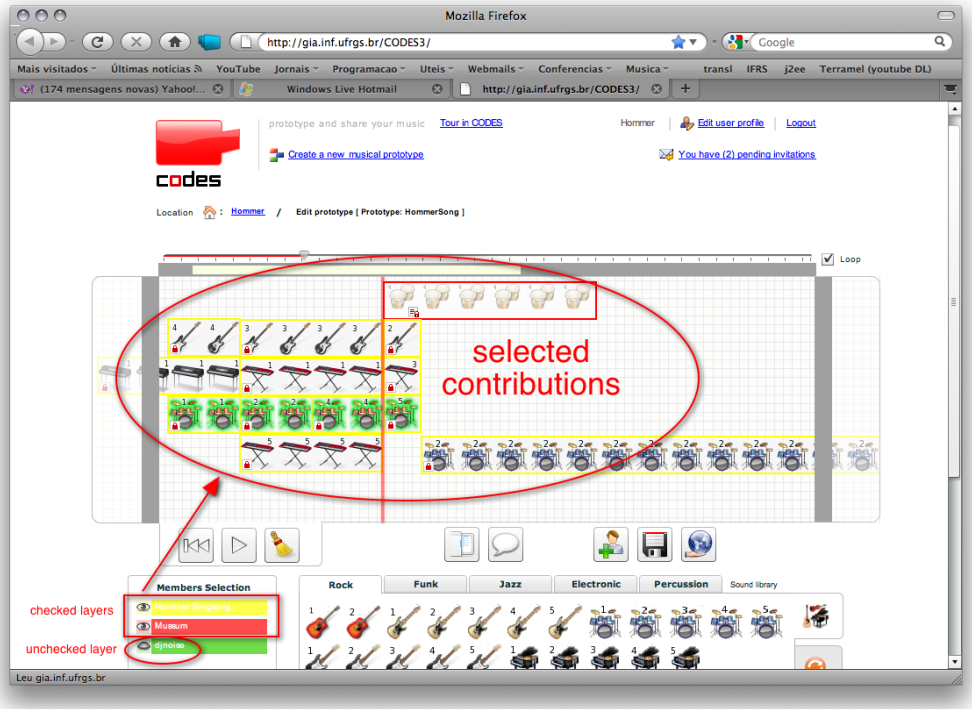


Figure 4.8: Listening to selected contributions

The “save” function is another option in which a cooperative edition can be

started. It means that one part of the musical prototype where the user is working on privately (which is invisible to others) can be set as "shared", enabling its edition by the partners.

CODES offers three different possibilities for saving a musical prototype:

1. Save as a *draft contribution*, which makes the contribution accessible and visible only to the user who did it;

2. Save as a *shared contribution*, which makes the contribution accessible and visible to the whole group with which the user is cooperating;

3. save as a *new prototyping version*, which creates a new version (duplicating the previous one) that groups all the contributions at the moment it was saved, and is also accessible to the whole group as an alternative option for those users who may want to start a new prototype at some point. By having different versions, the user will have the history of all changes in the prototype "at hand".

Figure 4.9 shows the options of the save window.

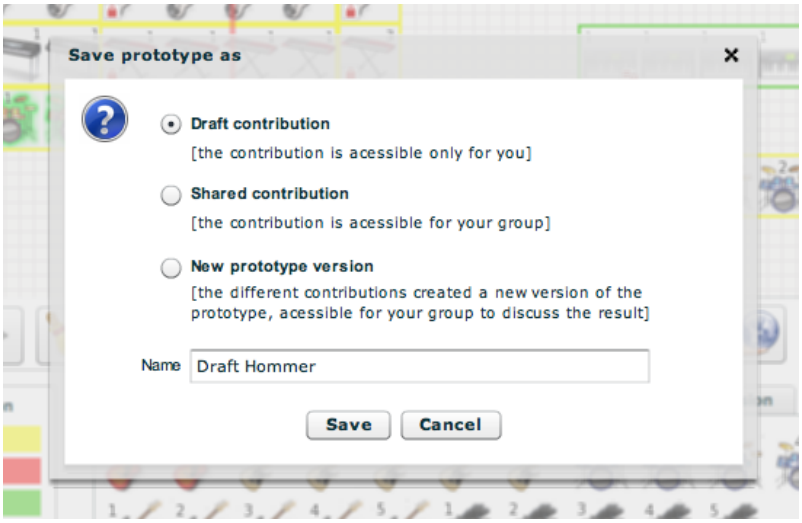


Figure 4.9: Options in the save window

Aspects related to MP Versions and Contributions are discussed further in the section 4.8.

Argumentation is another interaction mechanism and is part of the cooperative edition in CODES. The next section explains this mechanism.

4.7 Awareness Mechanisms

The concept of *awareness* has received a lot of attention in CSCW literature (see section 2.3.2) . In the context of CODES, the adopted notion of awareness is the understanding of the actions of other users, which provides for a user a context for his own actions.

CODES offers four kinds of awareness mechanisms:

- **MPR**, *music prototyping rationale* described initially in section 4.7.1, to allow users to know the reasons behind other members' actions;
- **Modification Marks**, to alert new events. See examples in Table 4.2;
- **Version Control with layers**, described in section 4.8, to keep an explicitly recorded track of the steps that led to the current MP state as shown in Figure 4.16 c);
- **Action logging**, to show the history of changes of the actions performed in the same MP.

This history of changes can be accessed in the action logging area, available in the tab on the right of the sound library. By selecting one entry (containing the user and the action) the awareness mechanism marks the corresponding changes with a yellow blurred background in the editing area. See these and other examples of these mechanisms in Figure 4.10.

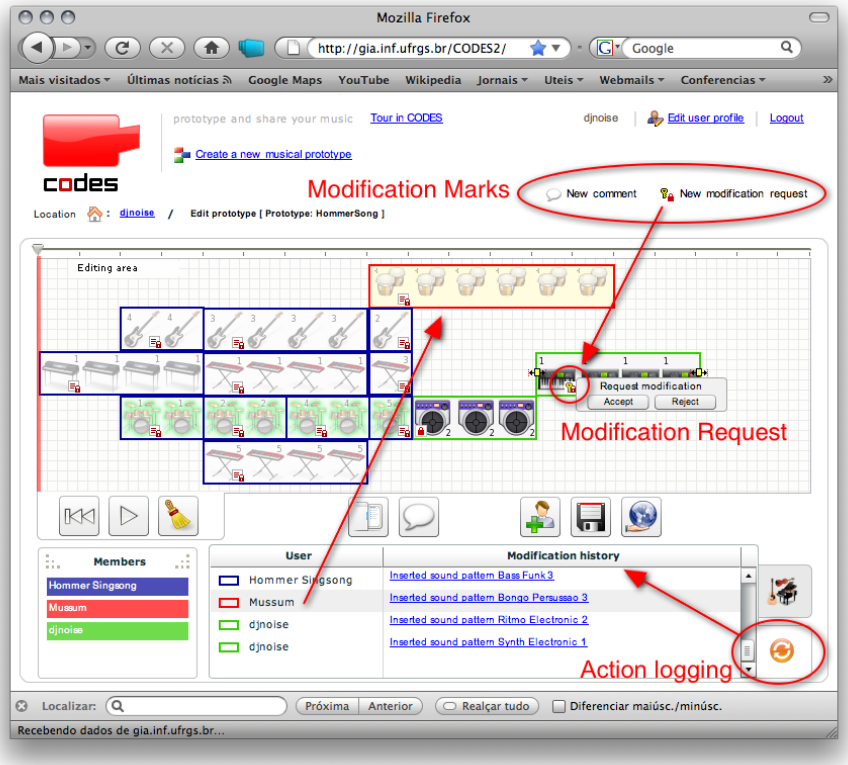


Figure 4.10: Example of awareness mechanisms of CODES

For each new action performed by a user, CODES registers it in a log and, eventually, marks it as a comment (with a "balloon" icon) or as a new contribution (with a "star" icon), depending on the situation.

By clicking on the icon, the user can retrieve information about the action from the log. Another important modification mark in CODES are the "padlocks" located in the sound patterns to be used in the modification requests, as shown in section

4.7.2. CODES marks with a different kind of padlock icon the sound patterns of an MP if a different user, from another layer, request permission to change it .

A brief list of the modification marks is shown in Table 4.2.

Table 4.2: Iconic marks for awareness in CODES

Icon	Status function	Context
	Agree	Position of a user sentence at the MPR
	Disagree	
	Idea	
	Important	
	My locked SP	Modification Requests status at the Editing Area
	My unlocked SP	
	Other's locked SP	
	Other's unlocked SP	
	Selected	Contributions visibility at the Editing Area
	Unselected	
	Publish	Publication Request status at the Intermediate level.
	Do not Publish	
	New comment	New action performed in the shared environment
	New action logging entry	
	New action performed	
	Pending invitation	

The first group represents the *positions* assumed in the "MPR mechanism". They mean that the user can position himself as "pro", "cons", suggest an "idea", or advise something "important", according to the argument that is being posted. They were detailed in section 4.7.1 with an example in Figure 4.12.

The second group is used by the *modification request mechanism* and shows to a user "my locked sound patterns", "my unlocked sound patterns", "others' locked sound patterns", and "others' unlocked sound patterns", respectively. Looking at these icons, the user may know what are the sound patterns that are available for others to edit, as well as what are the others' sound patterns available for edition. Using them, the user can set up (i.e. click on) the edition enabling it (with a green open padlock), or not (with a red closed padlock), or send a modification request if it is locked (others' sound pattern). The third group is used to inform which the user's contributions are selected in the editing area. The status may be switched by clicking over the user name (or its respective opened and closed eye at the bottom-left). The fourth group indicates the rate or amount of votes by users in order whether to publish (or not) the current musical prototyping. It will appear in the intermediate level, as the example in Figure 4.16.e) shows.

The last group indicates new actions in the environment, such as a new comment (i.e. balloon), a new action logging entry (circular arrows), a new change in the MP (with a star), and a pending invitation sent by other user (small envelop with an arrow). This new change can be a new sound pattern inserted or deleted, for example, and it will also appear at the intermediate level.

Each action performed triggers events in the shared musical piece; then a modification mark is shown to alert the users about such new actions.

Figure 4.11 shows an example of this last modification mark (pending invitation).

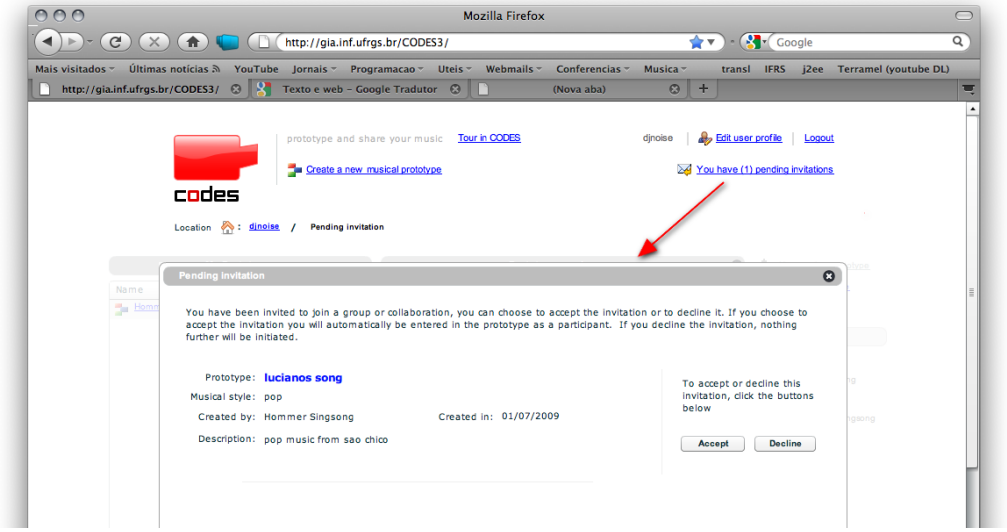


Figure 4.11: Accessing a pending invitation through its Iconic marks

The user can access the invitation and decide if accepts or decline it, by clicking directly on this modification mark.

4.7.1 Musical Prototyping Rationale

The *Music Prototyping Rationale* (MPR) mechanism is one of the original proposals of this work, based on the Design Rationale concept presented in section 2.2.3.

Each user may associate comments (i.e. an idea or an observation) and arguments (pro or cons) to any action on any prototype element. The arguments can be addressed to a specific user or to the whole group.

In CODES, the basic elements of the MPR are "issues", "positions" and "comments". Issues correspond to the decisions or actions that have been made, or states which have been reached during an MP creation and refinement. For example, issues may be "Removal of a sound pattern", "Pause or space inserted after the 4th sound pattern", etc. Issues are goal-motivated consensual choices, concerning alternatives of the course of action.

A Position is a statement or assertion concerning the issue. In the case of CODES, positions can be "pros", "cons", "idea", and "important". They are represented by the icons ☺ "smile", ☹ "sad", ☼ "light bulb", and an **!** "exclamation point" , respectively. See an example of CODES users in negotiation in Figure 4.12. These icons are merely informative. The intention is to give a clue at a glance to the readers about the user's opinion as a sort of speech act[1]. In this sense, an "idea" position should be related to the musical content while "important" should be related to other aspects of the MP or group. Regarding Figure 4.12, it is possible to see an idea sent by user *Hommer* (in the first line) suggesting to someone "to try another style", or mix different styles. Also, in the fourth line, the same user replies to an

[1] A speech act is an act that a speaker performs when making an utterance

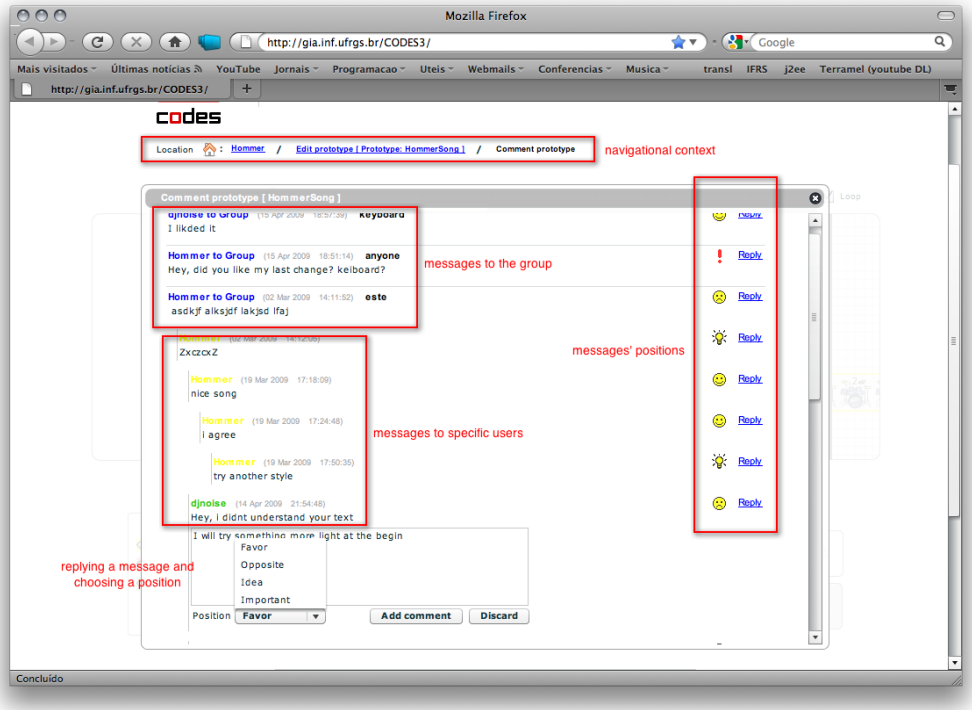


Figure 4.12: Music prototyping rationale in CODES

"important" post of the user *djnoise* with a new "idea" trying to solve a "musical problem".

Comments are made in order to agree with a specific course of action (comments "pro") or to express some objection (comments "cons"). Additionally, comments may express some suggestion, idea, question or generic observation about the issue. They are consensual explanations and not individual messages interchanged between actors, because they can be seen by all members of the group, and every decision or action may be linked to them.

As can be seen in Figure 4.12, the MPR of CODES uses a hierarchical structure to represent the reasoning of the users. Each entry in the structure may contain its author's username (User) and the content of the comment (Description).

Thus, a negotiation activity can be started and last from a few days up to several months. The MPR mechanism of CODES was designed to support this kind of asynchronous activity.

4.7.2 Modification Requests

Another possibility for CODES users to negotiate among themselves, in a cooperative edition for example, is to use the *modification request* mechanism. Modification request is a solicitation made in an MP position by some user aiming at changing other's (layer) sound patterns.

When someone contributes by adding a new sound pattern to an MP, it will be blocked by default for others users, with a blurred appearance. However, the author of the contribution can set its "changeable" status by clicking in the bottom-left corner of the sound pattern to unlock it, as shown in Figure 4.13.

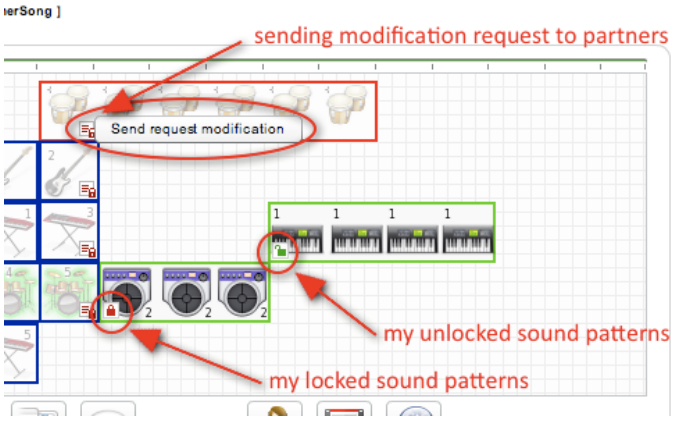


Figure 4.13: Excerpt of the Modification Request in CODES

This figure 4.13 shows a modification that is being requested in the blocked sound patterns of an other user. After the sound pattern is unlocked (by the owner), it will be available for other users, appearing not blurred anymore. This request is made directly on the locked sound pattern and a "modification mark" will appear for the sound pattern's owner, who may accept or decline the request.

Afterwards, the negotiation between these users can be complemented with arguments, their issues and positions, by using the *Musical Prototyping Rationale* mechanism presented in the section 4.7.1.

With this kind of control in CODES (enabling and disabling edition by others), the aesthetic intention of each user's authorship can be preserved. More about the authorship of contributions is detailed in the next section.

4.8 Versions: Preserving the Authorship

A technical-product oriented approach was adopted in the early implementation of CODES, including a tree-structure version control mechanism. It is traditionally used in the configuration and version management of software engineering projects (e.g. CVS), to avoid conflicts and inconsistencies among the several contributions of a group on the same shared project.

However, these mechanisms need some changes to incorporate specific characteristics for cooperative musical prototyping, as discussed in the section 4.3. As an example, supposing that three users (namely User1, User2 and User3) are cooperating in a shared musical prototype as shown in Figure 4.14.

Each node in the versioning tree (i.e. v0, v1, v2, v3) represents a version of the MP. Users can access any of the contributions to listen to, visualize, or edit it. When a new contribution (v4) is saved, a new node is created in the tree as a branch from the original primitive version (v1). Thus, the deeper is a node in the tree, the more refined would be the MP.

The advantage of this approach is that all the nodes are public and available for all users to make their contributions. However, one important disadvantage of this approach is the dependency between the user's contributions and the difficulty in keeping the original ideas, i.e., the "authorship" of each contribution.

Indeed, authorship is an important issue because, due to the non-technical nature

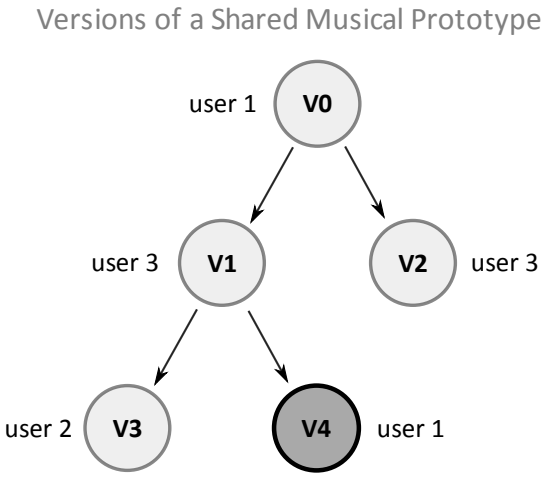


Figure 4.14: Shared Musical Prototype using a Versioning Tree

of CODES, there is no pre-defined product model to be followed, and the final result emerges from the creative and dynamic user interactions. If User3 wants to combine the contributions in node v2 with those in nodes v3 and v4, without including the contributions in node v1, it would be very difficult to do so, because these nodes (v3 and v4) inherit contributions from node v1. So, only those "added" contributions in nodes v3 and v4 would have to be "manually" identified and copied to node v2.

Besides, since those contributions were copied by User3, the authorship of User1 and User2 is lost in the new v2 version.

Likewise, User1 could delete all contributions by User3 when making version v4, which is another case of authorship conflict, since the contributions from User3 would then be lost in the MP, if the group continues the prototyping from node v4. To avoid this dependency between contributions and the possible authorship conflicts, CODES manages each user and his private contributions independently, as an individual layer like the layers-based structure shown at (SáNCHEZ; STRAZ-ZULLA; PAREDES 2008). See a representation of the layers in CODES in Figure 4.15.

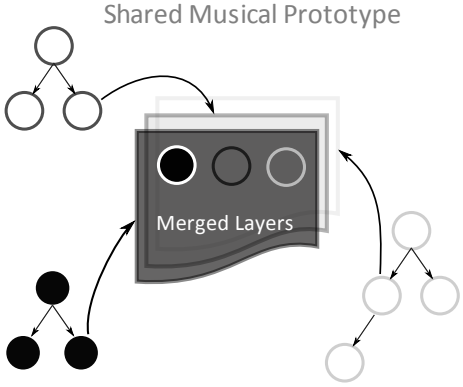


Figure 4.15: Shared Musical Prototype using Layers

In this approach, each layer represents one user's view, and the union of users' contributions (a combination of layers) results in a cooperative musical prototype version.

By using layers in the design, each user has his own tree structure, where each node represents a new contribution.

Other important characteristic of the layered approach is the independence between users' contributions, where it is possible to replace only one contribution without the need for parallel activities such as replication or sound pattern exclusion.

In fact, a new CODES level was created with this approach, which represents an "intermediate level". At this level, any user can browse between the contributions, independently of the creator, keeping the creator's original ideas and authorship. One example of outcome adopting the layered approach and how CODES organizes the users' prototype list is presented in Figure 4.16.

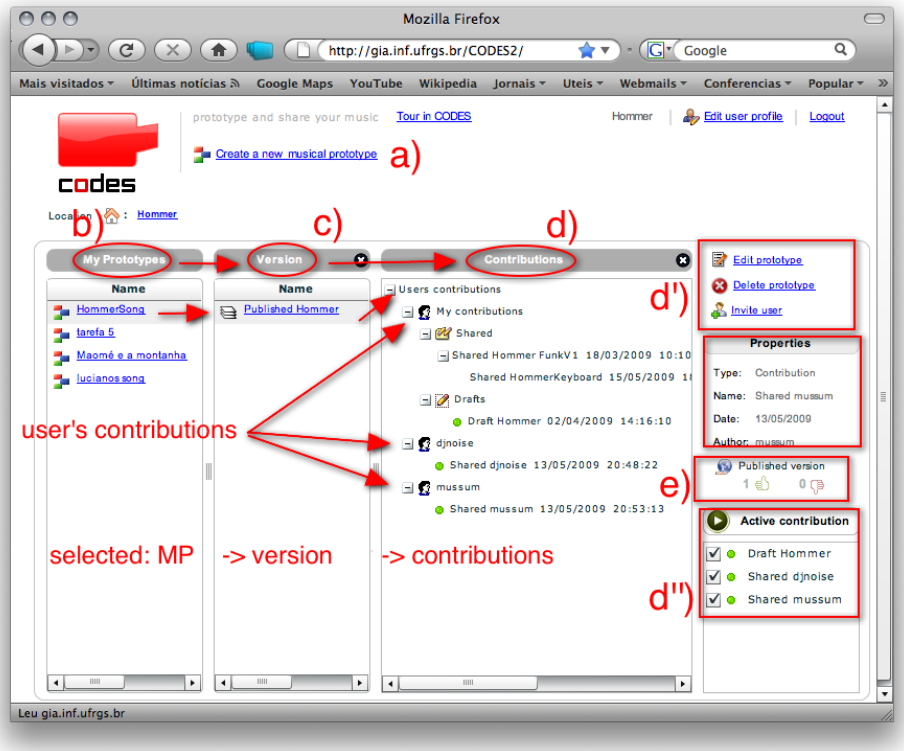


Figure 4.16: Layer Approach in the CODES user interface

CODES shows this screen to the members right after their login, also as a way of contextualizing MP's and actions.

Users can see their MP information in a kind of hierarchical structure by clicking in one of the My Prototypes list (Figure 4.16.b). CODES shows the nodes sequentially and dynamically, according to the user's selection. Each MP can have one or several versions (as shows the column "Versions" in Figure 4.16.c), which can also have one or several contributions (as in column "Contributions" at Figure 4.16.d). In fact, this screen is presented only with the column "My Prototypes". As users select their MP by clicking on it, CODES reveals the column Versions with all of

them. In the same way, by selecting some version, the next column contribution is presented with a list of all contributions and their properties ("draft" or "shared"). Such contributions can be selected and combined for listening (Figure 4.16.d") or edited (Figure 4.16.d').

If the user considers that in his opinion the MP should be published at CODES home page, he or she can trigger a *Publication Request* (shown in Figure 4.16.e). When clicking on this request, CODES indicates such intention to the whole group in a kind of rate, which can be increased or decreased and one can start a new negotiation about this subject.

As soon as new actions are performed, such as "edition", they are stored in the CODES database to be viewed by the group at this level as well. This "perception" entails the awareness mechanisms detailed in section 4.7.

4.9 Synthesis

This chapter has introduced the CODES support for a cooperative musical prototyping process in the Web, and a detailed description of the issues that this activity comprises. CODES provides support for cooperation over the Web for non-technical activities performed by novices, which implies a different approach for such a cooperation.

On one hand, novices do not have enough knowledge and confidence to create music. Under such circumstances, they need to experiment based on trial and error, which is the essence of the prototyping and the design. On the other hand, the non-technical activity such as the music created in CODES needs to support the unsystematic and opportunistic negotiation in order to assist novices and engage them and their interactions in the MP process.

With the support provided by CODES users may get better understanding of music creation activities by means of abstraction of concepts and alternative musical representation, as well as by interacting with other users, experienced or not, in order to get feedback.

The motivation for the Cooperative Musical Prototyping comes from the challenge of allowing novices create music by themselves. This activity can be performed by integrating Computer Music, HCI, and CSCW concepts. It has allowed the implementation of a solution that includes the cooperative and prototypical aspects in the music creation process of CODES, since it allows the manipulation of musical representation in the same way as in a prototypical process.

The expectation is that the cooperation through the CODES mechanisms can offer interesting possibilities for partners leveraging the user's interest in music, even the self-considered beginners.

Experiments and evaluations were planned to confirm this belief.

5 EXPERIMENTS AND EVALUATION

This chapter describes three experiments using CODES, and its evaluation. The goal was not only to get overall feedback (mainly subjective) from CODES users but also to investigate if the thesis proposals for novices engaged in non-technical cooperative music creation environment were accepted and approved be actual users.

The experiments were carried out following well-known evaluation methods from the HCI field, to obtain qualitative and quantitative results of the CODES environment and its functionalities.

CODES evaluation has been conducted through different usability evaluation methods, including Heuristic Evaluation (NIELSEN 1997, 1994a, b), and User Testing (RUBIN 1994).

Heuristic evaluation, for instance, is a widely accepted discount evaluation method for diagnosing potential usability problems in user interfaces.

User testing has involved actual users from different profiles, ages, and skills, interacting with the application to give feedback about user interface and interaction, user satisfaction, usability, and cooperative activities. In addition, it has included a set of tasks followed by the evaluation form to be filled out, according to a questionnaire with open and closed questions.

This chapter presents three experiments. Section 5.1 shows an experiment using versioning tree approach, section 5.2 details an experiment using layer approach, and section 5.2 shows the last experiment to collect corpus of requests aimed at assistance in CODES. Section 5.4 presents a comparison of CODES with related work, and section 5.5 summarizes the chapter.

5.1 Goal: MP using Versioning Tree Approach

Beyond verifying the usability problems, the underlying aim of this experiment was to check some cooperation issues with the versioning tree approach.

5.1.1 Participants

In this preliminary test, five individuals representative of the CODES typical users (3 male / 2 female) with ages from 19 to 35 years old, had to perform fifteen real tasks shown in 5.1.4.

Even though almost all participants were computer science skilled users (80%), they had no musical expertise and were using CODES for the first time. Their level of studies included 50% bachelors and 50% post-graduated students. Their previous experiences with musical software included only some simple activities related to

music sharing like search, download, and send music files over the internet.

5.1.2 Evaluation methodology

The experiment adopted User Testing method, and was conducted in the presence of one facilitator and one usability expert. He just read each task for the subjects and took notes of any problems found and any verbal comments from them. The subjects were instructed to talk about what they thought while interacting with CODES, in a well-known procedure adopted by the Thinking Aloud method (NIELSEN 1992).

After each session, users were also asked to fill out a questionnaire about their brief experience with CODES.

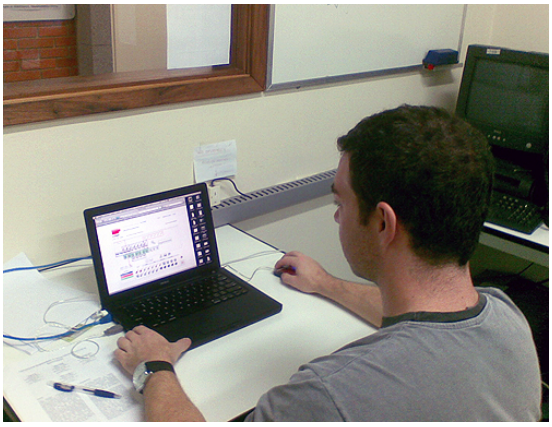
The average duration of each session was about 40 minutes.

5.1.3 Resources

Both the interactions and the comments were also recorded[1] by a video camera pointed at the computer screen, to allow further analysis.

To perform the cooperative activities, the experiment included another experienced user in CODES interacting with the system in a different room, with knowledge about the activities of the experiment. The Figure 5.1 shows an overview of a session being recorded and conducted by the observer and the expert as well.

a) Observer (left) and user (right)



b) The usability expert

Figure 5.1: A recorded session of user testing with CODES

The participants have used the resources of the computer science labs available at the Instituto de Informática - UFRGS. The hardware included PC and laptops with at least 1 GB of RAM, 120 GB of hard disk, Intel®CPU with 1.8 GHz. The software used was the Microsoft Internet Explorer and Firefox browsers running on Windows XP with broadband connection.

5.1.4 Tasks

The fifteen tasks were designed to simulate a scenario in which a novice user would learn how to create, edit and cooperate in a musical prototype. A description

[1] A demo session is available online at http://gia.inf.ufrgs.br/download/videos/ExpeVicky2.m4v (MPEG-4 QuickTime movie file - 347 MB)

of the tasks are listed as follows:

1. *Access and explore the CODES home page.*
   Goal: find out information at the public level about the musical prototypes, listen to 2 musical prototypes, and rate it.

2. *Get a CODES overview.*
   Goal: register in the system and login, and spend 5 minutes exploring the editing level, identify the sound library, the editing area, the preferred musical styles, the execution controls, and cooperation functions.

3. *Change the user profile.*
   Goal: identify the user profile area and change the color of the user.

4. *Create a new MP.*
   Goal: identify where to do it and fill out the information about the new MP.

5. *Edit the MP you have created.*
   Goal: identify areas at the P editing level, listen to the sound patterns of the sound library, chose a musical style, add and manipulate the sound patterns into the editing area, organize them and listen to the final music.

6. *Save the MP as Draft.*
   Goal: identify where to save and its options.

7. *Write something about the musical sequence you have created.*
   Goal: access the forum area and describe the last musical actions.

8. *Invite a user to cooperate with you.*
   Goal: access the user area, try to find a CODES user with the same style (the user B, in this case).

9. *Log out of the system.*
   Goal: log out for a while. At this moment some changes are done in the shared musical prototype by the user B, which also invites user A to cooperate in another MP.

10. *Check if there are pending invitations.*
    Goal: try to identify some modification mark and the new actions performed in the system.

11. *Edit and cooperate the MP you were invited.*
    Goal: identify and open the new shared MP, contribute, listen to it and write an argument.

12. *Save the MP as a shared version.*
    Goal: save a shared version.

13. *Log out of the system.*
    Goal: log out for a while. At this moment user B excludes the contributions of the user A and write arguments about it.

14. *Try to edit new contributions from the other user.*
    Goal: identify the changes and try to find the rationale for these actions.

15. *Identify and edit previous contributions.*
    Goal: identify where to visualize all versions, navigate and listen to them, choose one of them to edit, save a shared version and comment about it.

The next section presents some statistic data and discusses the results about this test.

5.1.5 Results

After performing the tasks, users filled out a form with open and closed questions. The open questionnaire posed eight questions concerning the Nielsen's heuristics[2], such as visibility, contextualization, control and freedom, feedback, flexibility, and the musical representation in CODES as well (see Appendix B, section 6.4).

To answer the eleven closed questions, one main question was made: *Do you agree with the following sentences?* The subjects should choose one of the five options: *Totally Agree, Agree, Neutral, Disagree, Totally Disagree.*

The average results extracted from this opinion poll for the eleven questions are presented in the next charts, and are followed by a discussion at the end.

1. The expressions and language used are clear and easy to understand.

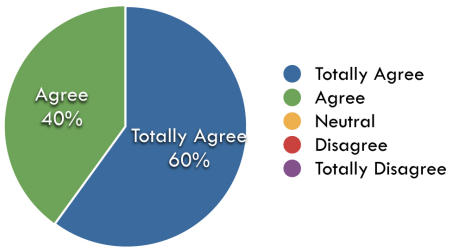


Figure 5.2: Opinion poll about the CODES language in experiment 1

2. It is easy to learn how to use the system.

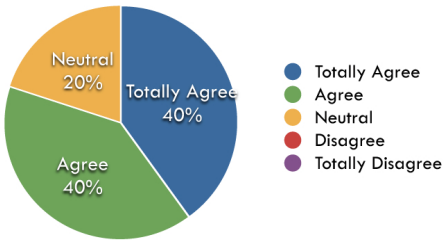


Figure 5.3: Opinion poll about the learnability of CODES in experiment 1

[2]http://www.useit.com/papers/heuristic/heuristic_list.html

3. After learning the system, it remains interesting and easy to use.

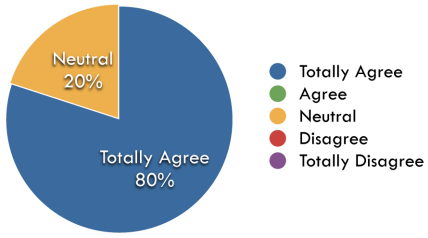


Figure 5.4: Opinion poll about the easiness and satisfaction of using CODES in experiment 1

4. The feedback from the system is adequately presented and easy to interpret.

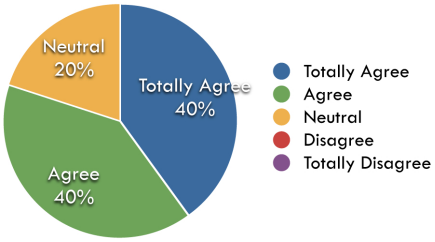


Figure 5.5: Opinion poll about the feedback from CODES in experiment 1

5. The look and feel of CODES is pleasant

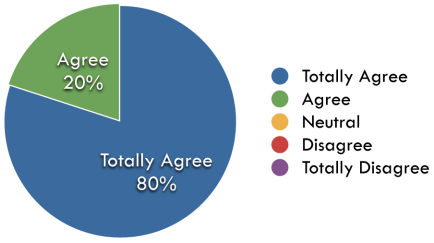


Figure 5.6: Opinion poll about the aesthetic design of CODES

6. The graphical sound representation in CODES helps to identify the sound content even without listening to it.

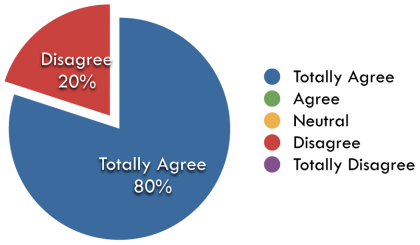


Figure 5.7: Opinion poll about alternative musical representation of CODES in experiment 1

7. Everybody may change all contributions.

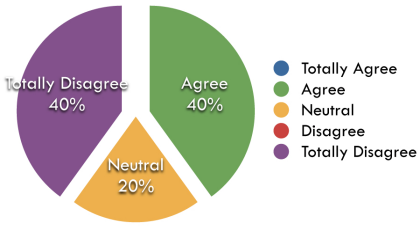


Figure 5.8: Opinion poll about the authorship of user contributions in experiment 1

8. The alerts allow easy understanding of the changes in the system

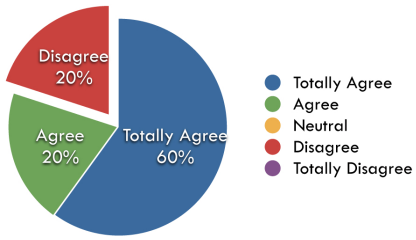


Figure 5.9: Opinion poll about awareness in CODES in experiment 1

9. The comments are useful for understanding the changes and versions of the music being created.

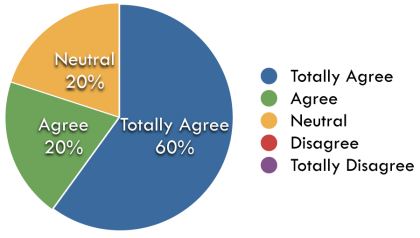


Figure 5.10: Opinion poll about the MPR of CODES in experiment 1

10. The contribution list represents an easy way of identify and understand the sequence of changes in the music

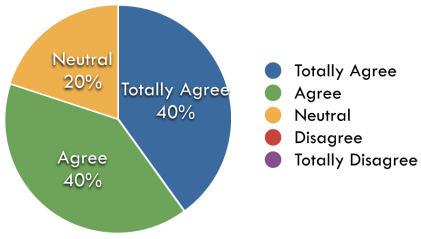


Figure 5.11: Opinion poll about the identification of changes in a shared MP in experiment 1

11. The cooperation mechanisms in CODES allow the group to achieve a consensual final outcome.

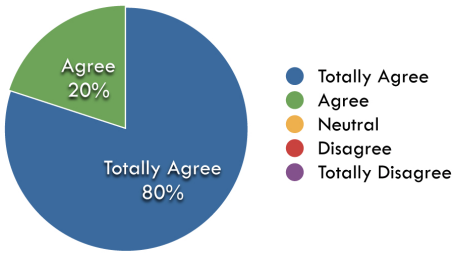


Figure 5.12: Opinion poll about the effectiveness of group mechanisms of CODES in experiment 1

On the overall, test results were favorable, however some inconsistencies need to be investigated more carefully.

Some users have detected important drawbacks concerning the system feedback, according to the following quotes extracted from the answers of the open questionnaire:

> "Sometimes, the system would give more feedback". "I do not know what is the session I am posting the comment". "I did not know why should I choose a color when registering myself in the system". "What means the icons in the editing area?"

Despite these negative points, most of them have assigned "totally agree" as shows the Figure 5.6.

Some divergence was found regarding Figure 5.8 and comparing it with a similar open question. Most of the subjects mentioned that they would like to block some contributions in order to preserve their previous works. Probably, this discrepancy not reflected in the chart is due to a bad formulation of the question, which was unclear for the users.

In the case of the awareness related to the *modification marks* shown in Figure 5.9, one of the subjects mentioned that it was not clear **when** there was invitation and **when** there was not.

According to one of the subjects, the graphical sound representation used in CODES do not help to identify the sounds, thus corresponding to the "disagree" part of Figure 5.7. In the later comments at the questionnaire, this subject argued as follows: "I disagree with it, because the notes and pitches can vary significantly

for the same musical instrument. The idea is very good, but the audio samples restrict to much the user..." he complemented, and mentioned some professional software for music edition like Guitar Pro[3].

This can be considered as an important point regarding expert-oriented systems. However, with regards the rest of users' opinions and considering the novice-oriented context of CODES, it may be reported as a discrete occurrence. Possibly, this one belongs to a group of more skilled users even though he has classified himself as a lay one in music.

A complete set of usability problems that have been discovered with the questionnaires is presented next, and discussed in order to rate the severity of each problem.

The following 0 to 4 Severity Rating (SR) scale described in table 5.1 was used to rate the severity[4] of usability problems.:

Table 5.1: Severity Rating Scale

SR	Description
0	I don't agree that this is a usability problem at all
1	Cosmetic problem only: does not need to be fixed unless extra time is available on the project
2	Minor usability problem: fixing this should be given low priority
3	Major usability problem: important to fix, so should be given high priority
4	Usability catastrophe: imperative to fix this before product can be released

Some users have answered a few of the open questions. The set of more significant problems extracted in the comments of the questionnaire are classified according to the following questions with the corresponding SR assigned by the usability expert:

1. *Are the expressions and the language used in CODES clear and easy to understand?*
   Answer of subject 1: "Yes, but it would be interesting to describe what each button does. They are not standard." **SR:2**.

2. *Is the look and feel of CODES pleasant?*
   Answer of subject 1: "Yes, but it would be interesting to put the list of users together with the editing area" **SR:2**.

3. *Is the feedback from the system adequately presented and easy to interpret*
   Answer of subject 1: "I don't know why I should choose a color when registering myself in the system" **SR:2**.
   Answer of subject 2: "Yes, but I thought that dragging the sound patterns outside of the editing area would delete them, but the system just hide them." **SR:3**.
   Answer of subject 3: "I don't know what the icons in the editing area mean" **SR:2**.

[3]A multitrack tablature editor available at http://www.guitar-pro.com/

[4]Severity Ratings for Usability Problems
available online at http://www.useit.com/papers/heuristic/severityrating.html

Answer of subject 4: "The system would inform about the required fields in the registering form" **SR:1**.

4. *Does the graphical sound representation in CODES help to identify the sound content even without listening to it?*
   Answer of subject 1: "I disagree with it, because the notes and pitches can vary significantly for the same musical instrument" **SR:0**.

5. *Did you get lost some moment or was there something you did not expect that happened?*
   Answer of subject 1: "I don't know what is the session I am posting my comment" **SR:3**.
   Answer of subject 2: "After editing my profile and saving it, the save button remains enabled" **SR:2**.
   Answer of subject 3: "After accepting the invitation, I would like to go directly to the musical prototype" **SR:3**.
   Answer of subject 4: "When replying to some message, I would like to see the original comment" **SR:3**.
   Answer of subject 5: "I can rate a published musical prototype as many time as I want." **SR:4**.

In contrast, most of the other opinions have met the project expectation, specially the following ones summarized both from the opinion poll and the open questionnaire:

a. The interface of CODES has been approved by all users regarding the usability aspects;

b. The system is easy to learn and use, keeping users connected after the first contact with the system;

c. CODES is accessible and intuitive, allowing novice users to experiment and collaborate in shared musical pieces without requiring musical skills;

d. The subjects have corroborated the initial expectation of the project in which users may want to control their contributions in the context of a group;

e. The MPR mechanism is useful for understanding the changes and versions of the music;

f. CODES awareness mechanisms are able to keep users informed about new actions performed in the system;

g. The users can easily concentrate themselves in the main task, and collaborate in the musical experiences, without getting lost;

h. The versioning tree approach is not adequate for keeping the users' original contributions in the context of the system;

From this full statistical corpus associated with the comments of the subjects, CODES seems to receive an initial approval of users after the test. Indeed, a significative number of suggestions were considered and implemented in the current version of CODES (as were presented in Chapter 4) .

5.2 Goal: MP using Layer Approach

Similarly as to the previous experiment, the underlying aim of this one was also to test the cooperation issues. The main goal was to check some questions pointed out in the previous experiment as well as to verify the new characteristics implemented according to the collected suggestions. Particularly, one of the aims was to check whether the users could understand and use the last version of the system with layer approach.

5.2.1 Participants

Similarly to the previous, this experiment was done with eleven participants. Most of the users were male, students with 25 years or more, and unskilled in musical software (except one student).

The observer and the CODES expert were not present at this time.

5.2.2 Evaluation Methodology

The methodology applied was user testing. This test was conducted by users participating asynchronously over a week. They should follow the instructions on two forms which were available online for guiding themselves during the test. One of the forms had the tasks which should be performed and the other form should be filled out after using CODES, and which contained objective and subjective questions.

5.2.3 Resources

In this test, the users have employed their own resources including as well PCs and laptops at home and work. The internet connection was at least 1 Mbps.

The hardware configuration was at least 1 GB of RAM, 80 GB of hard disk, Intel CPU with 1.4 GHz. The software included also the Microsoft Internet Explorer and Firefox browsers running on Windows XP.

5.2.4 Tasks

The sixteen tasks also tried to simulate a scenario in which a novice user using the system for the first time would learn how to create, edit and cooperate in a musical prototype. In order to do some comparisons, most of the tasks were the same as of the previous experiment (versioning tree) with little differences provided by the new implementation (layer approach).

A brief description of the tasks are listed as follows:

1. *Access and explore the CODES home page.*
   Goal: find out information and know more about the system by exploring the Tour in CODES.

2. *Get a CODES overview.*
   Goal: register in the system, login, and spend 5 minutes exploring the editing level; identify the sound library, the editing area, the preferred musical styles, the execution controls, and cooperation functions.

3. *Change the user profile.*
   Goal: identify the user profile area and change the color of the user.

4. *Create a new MP.*
   Goal: identify where to do it and fill out the information about the new MP.

5. *Edit the MP you have created.*
   Goal: identify areas at the P editing level, listen to the sound patterns of the sound library, choose a musical style, add and manipulate the sound patterns into the editing area, organize them and listen to the final music.

6. *Save the MP as Draft.*
   Goal: identify where to save and its options.

7. *Write something about the musical sequence you have created.*
   Goal: access the forum area and describe the last musical actions.

8. *Invite a user to cooperate with you.*
   Goal: access the user area, try to find a CODES user with the same style (the user B, in this case).

9. *Share the version you have created.*
   Goal: identify how to share an MP with other partners.

10. *Save a new version of your MP.*
    Goal: add some new sound patterns and save it also as a new shared version.

11. *Finding versions.*
    Goal: locate the new intermediate level and identify new versions you have created.

12. *Edit and combine different contributions.*
    Goal: enable different contributions at the intermediate level to be edited together.

13. *Identify the authors of the contributions.*
    Goal: Identify on the editing area all authors enabling and disabling their contributions.

14. *Argument about the changes you have made in the shared MP.*
    Goal: identify and use the MPR mechanism.

15. *Change others' contributions.*
    Goal: identify and use the modification request mechanism.

16. *Publish your MP in the CODES home page.*
    Goal: identify how to publish an MP.

5.2.5 Results

The results of the experiment 2 are presented in the next charts:

1. The expressions and the language used are clear and easy to understand.

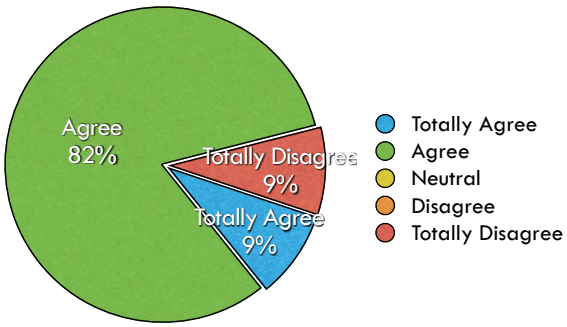


Figure 5.13: Opinion poll about the CODES language in experiment 2

2. It is easy to use the system.

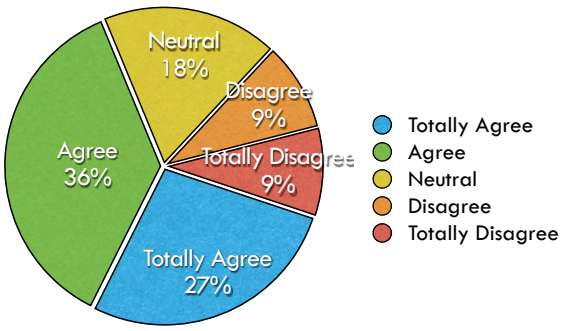


Figure 5.14: Opinion poll about the learnability of CODES in experiment 2

3. After learning, the system remains interesting and easy to use.

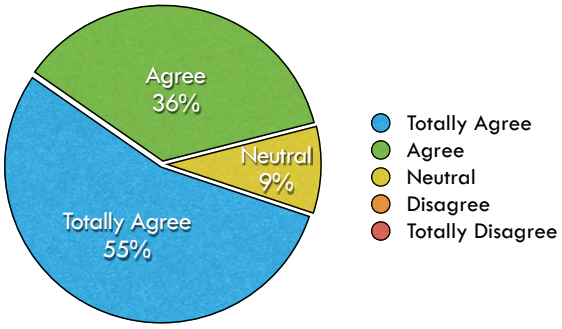


Figure 5.15: Opinion poll about the easiness of use and satisfaction using CODES in experiment 2

4. The feedback of the system is adequately presented and easy to interpret.

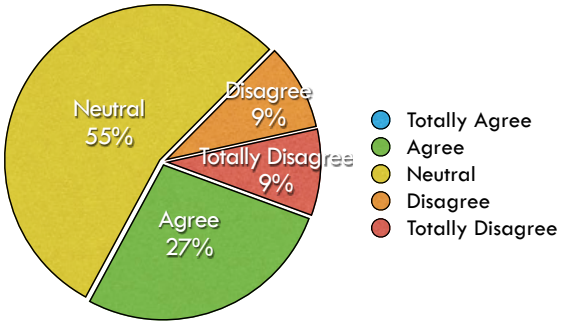


Figure 5.16: Opinion poll about the feedback in CODES in experiment 2

5. The look and feel of CODES is pleasant.

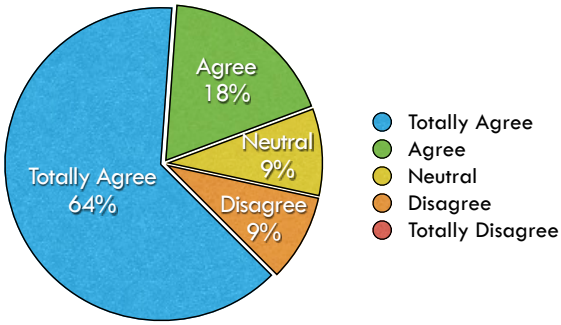


Figure 5.17: Opinion poll about aesthetic design of CODES in experiment 2

6. The graphical sound representation in CODES helps to identify the sound content even without listening to it.

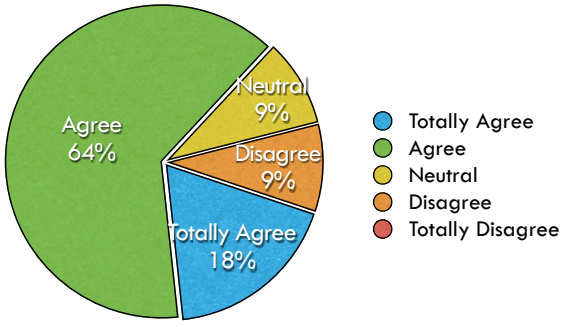


Figure 5.18: Opinion poll about alternative musical representation of CODES in experiment 2

7. The alerts (modification marks) allow easy understanding about changes in the system.

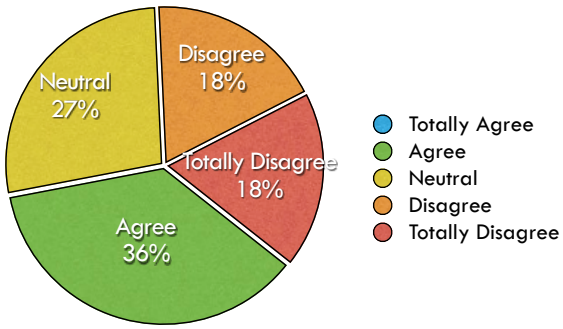


Figure 5.19: Opinion poll about awareness in CODES in experiment 2

8. The comments are useful for understanding the changes and versions of the music.

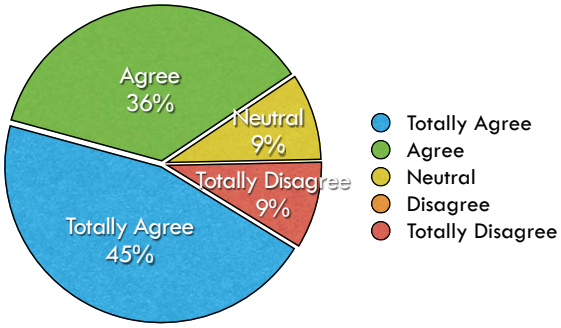


Figure 5.20: Opinion poll about the MPR of CODES in experiment 2

9. It is important to keep and show the different contributions.

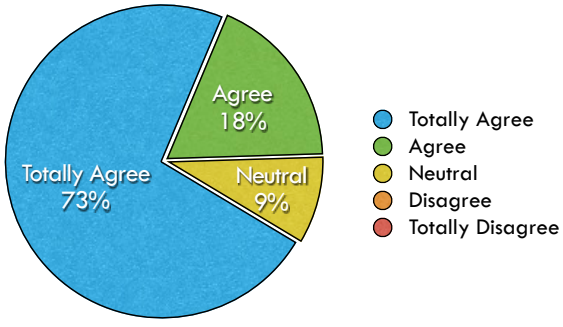


Figure 5.21: Opinion poll about the importance of keeping contributions in CODES - experiment 2

10. The contribution list represents an easy way of identifying and understanding the sequence of changes in the music.

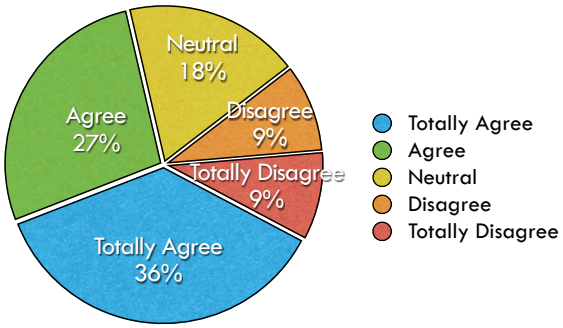


Figure 5.22: Opinion poll about the identification of changes in a shared MP in CODES - experiment 2

11. The possibility of locking the elements of the contributions is useful.

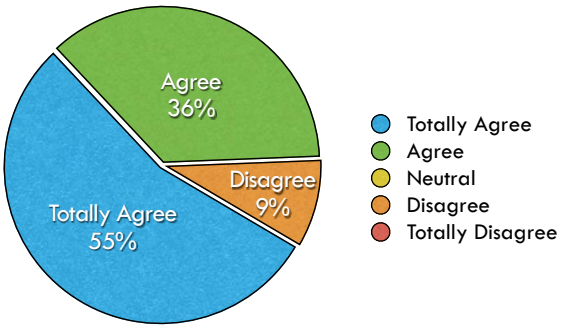


Figure 5.23: Opinion poll about locking musical elements in CODES in experiment 2

12. The possibility of merging different contributions is useful.

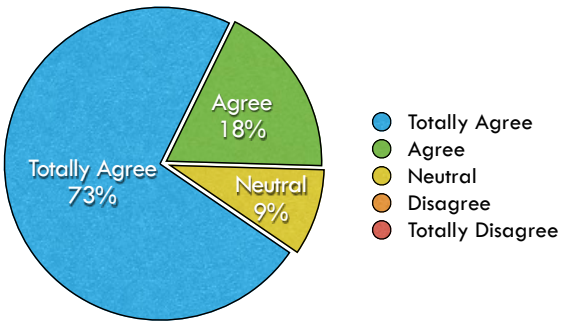


Figure 5.24: Opinion poll about merging musical elements in CODES in experiment 2

13. The CODES mechanisms help in the growth of user's interest in participating of the musical prototypes.

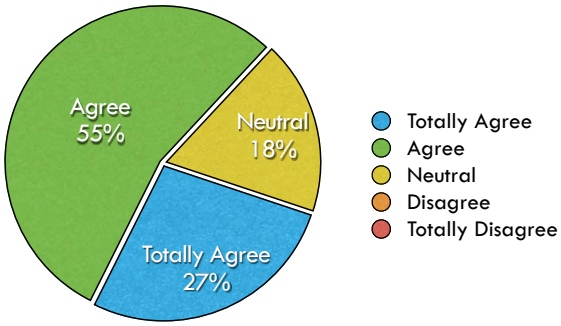


Figure 5.25: Opinion poll about the user intereset in CODES in experiment 2

14. The cooperation mechanisms in CODES allow the group to achieve a consensual final outcome.

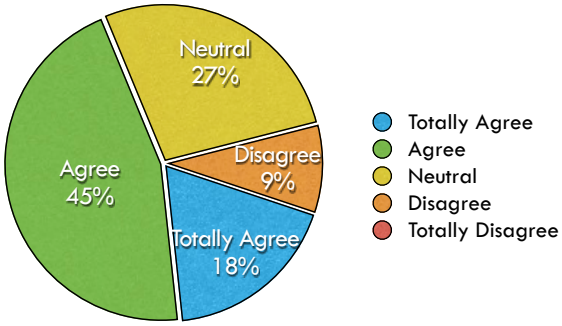


Figure 5.26: Opinion poll about the effectiveness of group mechanisms of CODES in experiment 2

Regarding the participants of this experiment (see section 5.2.1), there is one user with some skills in some music software. Since users could choose more than one option, the visualization of the number of each option selection was chosen to better illustrate the quantitative information.

Like in the previous experiment, results remained favorable overall. For those divergent or negative questions, the same Severity Rating used in the previous experiment was applied after the comments, in order to evaluate it. For instance, regarding the question about "system feedback" and the "alerts" - *modification marks* in Figures 5.16 and 5.19 respectively, it can be noted that the result differs from the previous experiment.

The charts themselves show divided opinions of users with most of them on the negative side.

The same 0 to 4 rating scale shown in table 5.1 and evaluator used before, in the first experiment, was again used to rate the severity of usability problems:

The users' comments are listed and evaluated as follows:

1. *Question about the "system feedback" and "alerts":*
   Answer of subject 1: "...I took some time to understand how to navigate

among the several functionalities ...” **SR:3**
Answer of subject 2: “It would be interesting to have text messages in a kind of status bar ...” **SR:1**
Answer of subject 3: “Sometimes the feedback takes so long after the action and sometime it fails...” **SR:2**
Answer of subject 4: “It lacks a message such as “successfully registered” locking the registration form.” **SR:3**
Answer of subject 5: “...I did not see any message after saving ...”, “...I did not get feedback of saving actions...” **SR:3**

2. *Question about unexpected actions or locations:*
Answer of subject 1: “ At the editing level, some sound patterns that were already excluded continued to play ...” **SR:4**
Answer of subject 2: “ I was trying to edit with another partner at the same time, and I didn't see her modifications. We have used chat (MSN) to inform each other of the modifications ...” **SR:0**
Answer of subject 3: “After changing my color, my data disappeared in the form.” **SR:1**
Answer of subject 4: “When I excluded a version of my prototype, the whole prototype was also deleted.” **SR:4**
Answer of subject 5: “After pressing save or update the system could close the dialog boxes. ” **SR:2**
Answer of subject 6: “ I expected to see my username at the CODES home page (my published prototypes) instead of my registered name. ” **SR:2**

3. *For each task, I can understand if it was completed or not regarding the system feedback*
Answer of the subject 1: “No, I didn't know if the prototype was really published in the CODES home page until check there.” **SR:3**
Answer of subject 2: “I didn't receive any feedback after saving a musical prototype.” **SR:3**

4. *Question about the graphical sound representation in CODES*
Answer of subject 1: “It could includes a label with a brief description of the sound or the musical instrument.” **SR:2**
Answer of subject 2: “It is ok, but due to their size, bass and guitar, for instance, it can be easily confused.” **SR:1**
Answer of subject 3: “As a suggestion, the tempo and intensity of each sound pattern should be considered as well.” **SR:1**

5. *Question about the increase of interaction provided by the mechanisms offered in CODES*
Answer of subject 1: “It was difficult to understand the versioning system, which is classified by user. In my opinion all contributions should be listed in reverse chronological order.” **SR:3**

About the answer in question number 2, the user suggests a chat tool for communication. CODES project was conceived initially to be asynchronous. However, such tool could be implemented in further versions of CODES with eventually synchronous characteristics. Suggestions such as “adding information about tempo and

amplitude" of sound patterns are out of the scope of this thesis, considering it is a musician-oriented concept and not for novices at all. However, it can be important to improve the flexibility and customization of the musical piece since it is available transparently for novices.

Other important suggestion that was mentioned is about the selection of the "list of favorites" sound patterns at the sound library. After listening several ones, it can be difficult come back to find a previous one.

Like the previous experiment, most of the opinions were positive and the approval shown previously was confirmed in this test. The suggestions will be taken into account to be implemented in the next versions of CODES.

5.3 Goal: Collect Corpus of Requests aimed at Assistance

This experiment aimed at collecting a small corpus of Natural Language Requests of Assistance uttered by novice users while interacting with the CODES framework to perform simple tasks. It was a very particular case that differs considerably from the previous one because of the specificity of the goal, which was developed in the context of an international cooperation project called PRAIA[5], in which the author of this thesis has participated and developed a doctoral stage at the LIMSI[6] lab, Paris-11 University, at Paris.

This work included the study of the LIMSI expertise on ACA (Assisting Conversational Agents), more specifically the DIVA (DOM Integrated Virtual Agents) developed by LIMSI, and the study of possibility of integration with the project of this thesis, on the cooperative music creation by ordinary users. Afterwards, a second step included the current experiment described herein to evaluate the CODIVA, which comprises the integration of CODES with the DIVA agent. A preliminary result of such an integration (under development) can be accessed in the CODIVA Web page[7].

5.3.1 Participants

The sessions experiment involved 12 subjects (4 male / 8 female) between 23 and 35 years old, most of them MSc and PhD students (from diverse domains: architecture, law, cooking, acrobatic dance, urbanism, history, psychology, physics - and from diverse countries: Mexico, France, Brazil, and Germany). To keep the homogeneity of the sample group and, hence, of the collected corpus, no subject had any background knowledge, either in computer science (beyond surfing the web) nor in music (theory or practice), and may thus be considered as novices in both domains.

5.3.2 Evaluation Methodology

In this experiment, all subjects were also told to "think aloud": that is, to express orally the questions they could have while performing the tasks, as if there was an expert "friend behind their shoulder" (CAPOBIANCO; CARBONELL 2001), played by a Wizard of Oz (WoZ) operator.

[5]Pedagogical Rational and Affective Intelligent Agents: at <http://gia.inf.ufrgs.br/praia>

[6]Computer Sciences Laboratory for Mechanics and Engineering Sciences: at <http://www.limsi.fr>

[7]Availabe online at <http://gia.inf.ufrgs.br/lcmCODES/diva/music/music.main.htm>

This choice was taken to let them be as free as possible in the utterance of their questions.

Users should learn how to edit a music prototype, which included the three following stages: a) to choose a music style together with their related sound patterns; b) to create and play a sequence of patterns; and c) to edit the sequence they have created.

These three stages were decomposed into five different tasks to be performed on the Web page of the music prototype editing level. They represent the steps that users typically follow in order to build an actual music prototype in the CODES system.

To this purpose, five simple tasks to be performed by the users were defined, and are described in section 5.3.4.

Interactions were recorded because, even being briefed about the think-aloud protocol, subjects often uttered generic or vague questions with anaphoras and coverbal deictic gestures, such as: "why does it not work?" - which could be a problem during the subsequent analysis phase.

Figure 5.27 shows a photo of a recorded session.

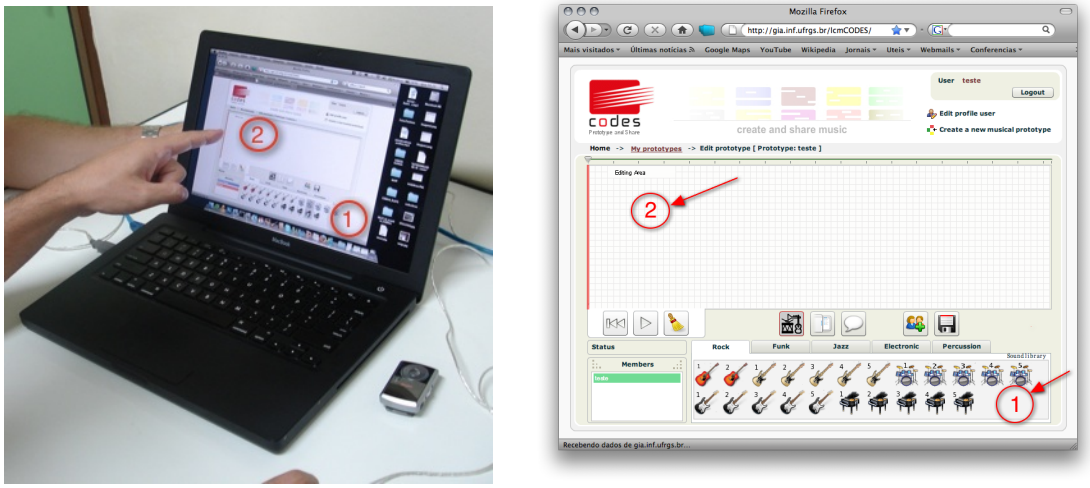


Figure 5.27: A screenshot of a video-recorded session

a) on the left, a subject is asking: "Should I put it there?" while dragging a selected pattern icon (1) = "it" and pointing his finger on the editing area (2) = "there" ; b) on the right, the window of the CODES Musical Prototype Editing Level.

Thus, by watching back and analyzing the recorded videos, it was able to resolve many issues and identify the objects referred to in the requests. After the experiment, the subjects had to fill out a questionnaire.

5.3.3 Resources

Video cameras and mp3 players were used to register both the interactions and the questions uttered. Most of the interactions were done with a Macbook laptop with 2GB of RAM, 250 GB of HD and 2.4 GHz of Intel CPU. The software included the FireFox browser version 3.0, running on Mac OS X with broadband connection with at least 1Mbps.

5.3.4 Tasks

1. *Choosing: a) a favorite musical style, b) three favorite sound patterns from that musical style.*
   Goal: to identify the sound library, the musical styles available and their sound patterns, selecting three of them. The idea is to check if the users could identify and navigate through the different musical styles described in the tabs (Rock, Funk, Jazz, etc.) and to listen to different sound patterns of each style.

2. *Putting the sound pattern in the editing area to compose a musical sequence.*
   Goal: to add the sound patterns in the editing area to create musical sequences. The idea is to check if the users are able to drag the icons from the sound library and to drop them onto the editing area

3. *Listening to the sequence they have created.*
   The goal is to check if the users identify and associate the execution control buttons with the editing area. When clicked on, the "Play button" changes itself into a "Pause button" and is filled with a gray color in order to give feedback to the user where to stop listening.

4. *Deleting one sound pattern they did not like, and changing to another they think sounds better.* Goal: to check if the user understands how to exclude a sound pattern and to replace it by another one. The metaphor adopted here is that of a broom button to enable the erase function. A flip-flop control was used: the user has to press the "broom icon" (button) to activate the erase mode, and to click again to stop the erase mode.

5. *Creating a music prototype with at least five sound patterns, in which some sound patterns (three at least) must be played at the same time.*
   Goal: In this task, it is implicit the notion of the musical track that allows users to test the combination of samples by playing them at the same time. The users should put the sound patterns under or above each other and listen to the result of that combination: when the red vertical timeline on the editing area, which runs when the play button is pressed, touches the icons their respective sounds are played.

5.3.5 Results

During this experiment, all subjects asked for assistance from the CODES expert. They were then given the best possible help information and, consequently, all subjects completed all the tasks successfully. This resulted in a corpus of 115 Natural Language requests, acquired over a month.

Table 5.2 displays an excerpt of 30 verbal utterances, transcribed off-session from the audio data.

Using the recorded sessions, utterances have been associated with additional contextual information, thus enabling the analyst to solve anaphora (e.g. "it" pronouns - in lines 1, 5, 6, 9...) and indexical items, (e.g. "this" - in line 15, 18).

Although if a few utterances are not assistance requests regarding the structure or the functioning of the application (e.g. lines 20 expresses a subject preference), most actually are.

The requests have been collected in Portuguese, Spanish, and English, and in the first two cases then translated into English for further processing.

Table 5.2: An excerpt of 30 questions extracted from the CODES corpus

1	Why doesn't it make any sound?
2	If they are not well aligned they will play?
3	Can I use the same sound pattern?
4	Can I repeat?
5	How can I stop it?
6	How should I listen to it?
7	How can I turn off the broom?
8	How can I come back?
9	Doesn't it run?
10	Are the sound patterns the instruments?
11	What do you mean with 3 sound patterns?
12	Should I choose only 3?
13	To play together should I put them behind one another?
14	Where is the login area?
15	Is the editing area this blank area?
16	I don't know where is the editing area?
17	Are the musical styles: rock, funk, jazz..?
18	Is it this way?
19	3 instruments?
20	I like pianos!
21	How can I undo the broom?
22	Why does it change?
23	Are they the sound or the instruments?
24	Is it necessary to click on play to listen?
25	Should I record before?
26	Where can I play?
27	How can I play the whole sequence?
28	Can I put them back?
29	How do I know the sound patterns?
30	I don't understand why when I click in the n.1, the n.5 appears?

5.3.6 Quantitative Analysis of the Assistance Turns

It was observed that assistance was often provided by answering a single question, when the CODES expert's answer being enough for the subject to be able to continue the task at hand. Thus, each assistance turn corresponds to an independent pair of question/answer and not to a real dialogue (in the worst case, there sometimes was a "re-phrasing dialogue" where some questions were asked twice or three times with some linguistic variations).

Subject Loquacity. The general data from the assistance turns related to the five tasks is represented in Figure 5.28, with subjects sorted from the most loquacious (i.e. help-seeking) to the least loquacious ones.

Note that for each task, in the group of subjects, there is a large inter-subject variation. It is also shown in figure 5a, with a range factor of 4 between Julia and Julieta. This may be proportional to the familiarity with other Web-based applications, not necessarily related to music.

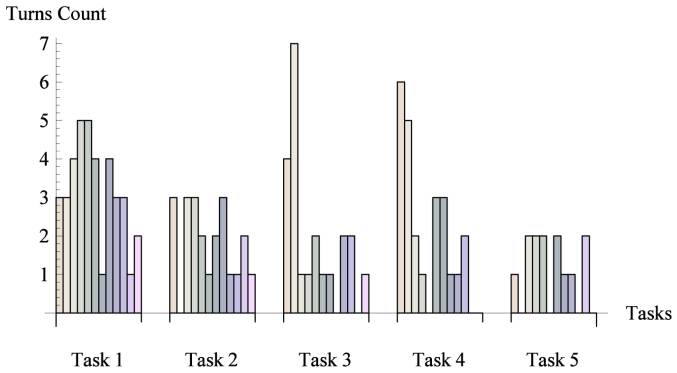


Figure 5.28: Overview of turn counts per task and subject (1 bar = 1 subject). Subjects are sorted by their global loquacity (from maximum to minimum)

The Novice Effect. In Figure 5.29a, it is possible to see that whereas the five tasks were defined with increasing complexity, the total amount of turns indicates that while the subjects progress from task 1 to task 5, they tend to require less assistance. This is emphasized by the turn count of task 1 exhibiting a peak of

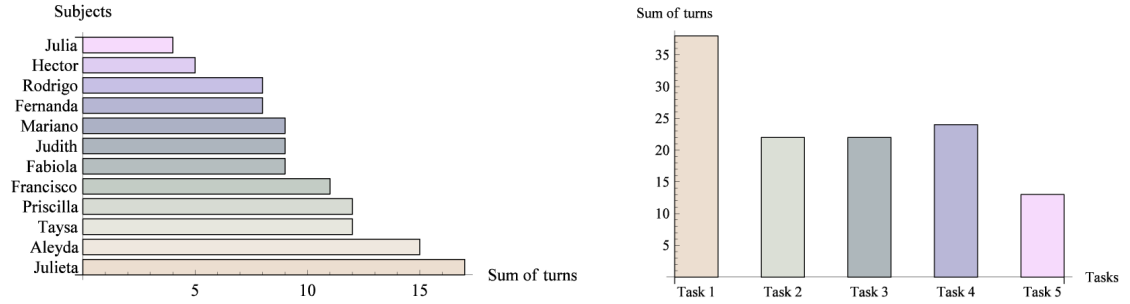


Figure 5.29: a) on the left, the loquacity of the subjects: the sum of turn counts on all tasks for each subject; b) on the right, the need for assistance on the tasks: the sum of turn counts over all subjects for each task.

assistance turns.

Analyzing the interrelationships among the tasks, it is possible to suppose that some questions related to one action performed, in relation to a given task, were not repeated when a similar action was performed in another task execution.

This clearly shows a so-called "novice effect" where assistance is mostly needed when the users enter the application, and tend to decrease even when the users are involved in more complex tasks.

References. Referential objects are "things" that the users categorize "in their mind" and refer to in their linguistic expressions. They can be actual GUI objects (buttons, browser, window...), screen areas and panels, icons representing application items like the musical prototypes (MP) or the sound patterns (SP); other significant elements referenced in the corpus are the broom button, numbers, and application-specific concepts like musical-styles or musical-instruments (for simplicity, they are all collected in a single class called MUSIC).

Their notation, definition and corpus distribution is given in Table 5.3:

Table 5.3: Notation, definition and corpus distribution of collected requests

Reference	Description	% Corpus
SP	sound pattern	59%
EA	editing area	33%
SL	sound library	16%
GUI	GUI objects = keyboard, button...	10%
MUSIC	music-related objects = styles, instruments...	9%
Broom	button used to delete the sound patterns	8%
SPL	sound pattern library	5%
NUM	any integer number	4%
MP	musical prototype	4%
SPEA	sound pattern editing area	2%

This experiment has presented briefly an approach to facilitate the design of Assisting Conversational Agents dedicated to the function of Assistance to ordinary people interacting in Natural Language with assisting agents in CODES. In this particular context, the key issue is the feasibility and the cost effectiveness of the adaptation of the assisting agent to CODES. The approach is based on a) the collection of a specific corpus of textual requests with novice subjects asking help while performing predefined tasks; and b) the elicitation of the specific linguistic entities through an analysis of the collected data. Detailed explanation about the procedures of this experiment can be found in (MILETTO et al. 2009).

5.4 Comparing CODES to Related Work

As part of the evaluation, this section aims at comparing CODES with other related work presented in section 2.5. For this, all the issues presented in chapters 2 and 4, the fundamentals and CODES proposals respectively, are now related and discussed in this section. Table 5.4 shows the same classification criteria presented in chapter 2, but with CODES included in the last column to highlight the differences.

Some important comparisons can be made regarding the data of table 5.4.

Table 5.4: Comparing the Related Systems with CODES

	JamSpace	Pitch Web	EduMusical	WebDrum	Daisyphone	PSO	CODES
Sound format	MIDI						mp3
Musical process	jam session	performance	composition	performance	performance	performance	experimentation performance
Sound representation	scratch tracks	geometric shapes	piano roll	grid	dots, circle	bouncing balls	iconic sounds
Sound Exportation	-	-	.mus	-	-	-	mp3
Architecture	Client-server						
Access	restrict	free	restrict	free	free	free	free
Interaction	synchronous, asynchronous	synchronous	synchronous, asynchronous	synchronous	synchronous	synchronous	asynchronous
Communication tools	-	chat	chat	chat	shared screen (draw)	-	mail / MPR
Persistence	-	-	yes	-	yes	-	yes
Group memory	-	-	-	-	-	-	yes
Awareness	-	-	-	-	-	-	yes
Interaction trace	-	-	-	-	-	-	yes
Argumentation	-	-	-	-	-	-	yes
Authorship	-	-	-	-	-	-	yes
Requirements to run	software hardware LAN	ShockWave, QuickTime, Beatnick	Java	JSyn	Java, Beatnick	Java	Flash Player
Target public	musicians, novices	composers, novices	OSESP students	novices	novices	novices	novices

With respect to the "musical process" criterion, even though situating itself as an entertainment system, CODES can be used also for "performance" and "musical creation" by means of experimentation with sound patterns. Since users can add and remove sound patterns while listening to the musical prototype, CODES enables some characteristics of live performance such as interacting with the musical piece while it plays.

Regarding the "sound format", it has been decided that CODES would accept mp3 files to take advantage of Flex engines for sound manipulation and also for audio quality purposes, which would produce more engaging audio results to stimulate novice interactions. CODES, like other related systems, uses a graphical "sound representation", and in addition it uses an iconic representation to try to give some clue about the sound which the image represents. Similar to most systems, CODES uses client-server as Web "architecture" to allow free access for Web users. Considering that users access the system in different times and can have musical ideas at any time, CODES has adopted asynchronous "interaction" and communication. Nevertheless, synchronism can offer interesting possibilities and can be taken into account in future perspectives. To access and "run" CODES, it is required the Flash Player plugin, that is a widely distributed proprietary multimedia and application player, sometimes, built into the browser. Whereas few systems use chat or drawing in the screen as "communication tool", CODES, in contrast, uses e-mail and its original mechanism *music prototyping rationale* as a permanent argumentation tool, in which users can understand positions and reasons of other users. Arguments in CODES are of consensual explanation, not an individual message interchanged between actors. Decisions are goal-motivated consensual choices, concerning alternatives of the action course. Every decision or action may be linked to arguments (pro or contra).

Clearly, the great difference between CODES and related system is in the novice-orientation characteristics supported by the CSCW mechanisms.

Indeed, most of other systems do not consider "persistence", "group memory",

"awareness", "interaction trace", "argumentation", and "authorship" as important aspects to engage users in the collective musical activities.

Considering the *Cooperative musical prototyping* as a process that involves people creating groups and working together on an MP as a shared workspace, this research illustrates how these criteria should be taken into account in the design.

The next section discusses the overall analysis of the results.

5.5 Synthesis

This chapter has presented some experiments carried out in the last year of the thesis project. Different subjects were tested in order to get an overall feedback from the users regarding usability, accessibility, and cooperative issues.

The results are promising. In the first experiment, the tests aimed at discovering the interface and interaction drawbacks and also was focused on the need of keeping authors' original contributions. Figure 6.8 shows the sum of the indices of the opinion poll about the system, obtained from the questionnaire presented in section 5.1.5.

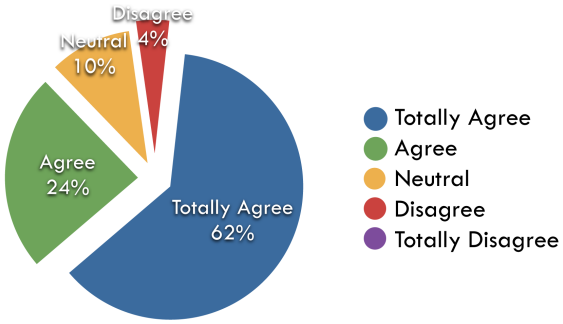


Figure 5.30: CODES General approval

CODES was designed to be as simple as possible for novices engaging in collective musical experiments over the Web. Considering that most sentences are affirmatives during the experiments, this research work considers that this goal has been successfully achieved.

It is necessary to know the target user as presented in section 2.2.2 (a novice in music in the case of this research) in order to design CODES to support them. Based on this idea and after these experiments a set of novice-oriented requirements, described in section 4.4, was presented as part of CODES design.

Regarding the related work presented in the section 2, one of the main advantages of CODES is on the support for the effective cooperation of novices.

The works are networked systems basically for technical reasons (much as easy access, publishing material, server support, maintenance, etc.); while this research considers, besides these reasons, that the Web is the best way for users to cooperate themselves, since they are novices in music.

In a cooperative system such as CODES situated in the domain of Arts, considered as a non-technical domain (see section 4.3 for detailed explanation), the use of DR is imperative since users do not know the final result of heir work. This research proposes an adaptation of this concept, herein defined as *Music Prototyp-*

ing Rationale, presented in section 4.7.1, associating it with novice-oriented specific requirements for musical interaction.

CODES is located in the bottom-right quadrant of this figure, even though the coordination role may not be as explicit as presented in Table 4.1. Based on the 4W1H questions presented in section 2.3.2, an awareness mechanism is proposed aiming to contextualize CODES users in their activities (see details in section 4.7).

CODES relies on the focus of the process - which is adequate for novices - and not on the aesthetic and final result.

The experiments were intended to be developed in a very restricted context; however, extra data has appeared in the body of overall results. Even when knowing these terms and conditions, some users felt the need for engaging partners motivated by themselves after trying CODES. Thus, is possible to infer that a) the general purpose of CMP is well accepted, i.e., users (specially the novices) want to create music, independently of their background; b) even presenting yet some initial difficulties due to the learning curve, the CODES supports for CMP is good enough to stimulate users to invite their friends as partners and to make users feel motivated in using CODES for enhancing and sharing their musical experiments, and c) more generally, people need to find new ways of expression. CODES is just another (good) way to allow this.

6 CONCLUSION AND FUTURE WORK

This final chapter presents a summary of the contributions of this thesis in section 6.1 and how each of them was evaluated; section 6.2 discusses some limitations which were recognized in the research, and section 6.17 proposes some perspectives of this research.

Through the development of the CODES project it was possible to overcome those obstacles presented in chapter 1, in the following way:

a. Develop a novice-oriented user interface, with HCI and CSCW concepts to make possible the prototyping and cooperating activities. Thus, the barrier of the expert-oriented musical technology interfaces was eliminated (presented in section 4.4).

b. Use the Internet and recommendations about standard architectures, as much as possible, to provide universal access (presented in section 3.4).

c. Use alternative musical representations to break the barrier of musical notation described in section 4.6.2.

d. Create a multilevel environment to enable the use of PCs as a virtual musical instrument (shown in section 3.6) to eliminate the barrier of own and carry a real one.

e. Proposing a direct manipulation process to systematize the musical experimentation, both individual and collective (presented in section 4.5).

f. Adapt the CSCW and HCI concepts like “awareness” and “design rationale”, respectively, to propose specific mechanisms for novices to perform effective interaction and cooperation with each other, to break too the barriers of visualization, argumentation, and understanding of musical actions (see sections 4.7.1, 4.7.2, and 4.7).

g. Creating a mechanism for versioning management in order to keep the authors' original contributions as presented in section 4.8.

As a result, this thesis provides a set of contributions summarized in the next section.

6.1 Summary of Contributions

All of the contributions of this thesis were aimed at solving the gaps and barriers described in section 1.1, and compared with related work synthesis presented in section 2.

The general contribution of this work is a new viewpoint concerning how should a technological infrastructure be aimed at providing support for music creation by novices. For the viewpoint proposed and developed in this thesis - that the music creation should be both cooperative and prototypical - CODES is a Web-based environment built as a proof-of-concept to provide support and to allow experimentation with these aspects and several related ideas.

CODES includes an approach to allow effective cooperation among novice users who require adequate support for producing and discussing their music.

The investigations in the course of this work have produced some relevant contributions, namely:

a) **The definition and application of two main principles to encourage novice-oriented music creation activities: i) they should be prototypical and ii) they should be cooperative**.
A *prototypical music creation* process means that novices can draft simple musical pieces, called here *Musical Prototypes (MPs)* which can be manipulated in a cyclical refinement process from the initial musical sketch until a final stage is reached. This process clearly resembles prototyping cycles adopted in industry and in incremental software development. Since music creation is a (music) design activity, it seems natural to adopt a prototypical process usually employed in design activities. If novices do not have enough knowledge and confidence to create music, they need to experiment based on trial and error, which is the essence of prototyping and of the design.

   In a *cooperative music creation* process, the refinement of an initial musical idea is the consequence of a collaboration of their authors and so, their partners. All members of a group (in fact, a social nework built by explicit invitation) will be cooperating until a consensual stage of the MP be reached. This process is noticeably a particular kind of Human Centered Collaborative Design where the result of the design is an MP. The non-technical activity such as the music created in CODES needs to support the unsystematic and opportunistic negotiation required in order to assist novices and to engage them and their interactions in the MP process.

b) **Integration and adaptation of Computer Music, HCI and CSCW concepts, methods, and techniques**.
The adoption of, for example, the "awareness" concept from CSCW and the "design rationale" from HCI, which are not usually found in computer music works but are very important to CODES; Such concepts were applied and adapted in the *Group Interaction Mechanisms* of CODES, as described in section 4.6. They allowed the proposition of concepts such as the *music prototyping rationale*, introduced here in this work as a mechanism for argumentation activities for sharing knowledge.

c) **Definition of concepts, features and system requirements for novices in music experimentation**.

Examples extracted from the CODES project, in which the user interface design has a totally novice-oriented focus, are: the graphic musical representation proposal, for instance, which is a high level representation scheme that allows direct manipulation of sound samples by lay people in music. This proposal overcomes the need for knowing musical notation in order to experiment with musical creation. It was positively evaluated by users in the very first experiment shown in the Chapter 5;

d) **Proposition of the *Cooperative Musical Prototyping* concept (CMP)**.
CMP is indeed a simple and straightforward process for music prototype creation. It is iterative, incremental and evolutionary, since an initial musical idea (first version of an MP) is produced and refined through a number of stages up to the final version. All those characteristics presented in Chapter 4 become transparent for novices overcoming their own barrier and get to know how to play an actual musical instrument. With the usual interaction metaphors like "drag and drop", novices can manipulate musical elements or building blocks defined here as *sound patterns*, to create their musical content, by inviting others to cooperate and share it in a public space. Moreover, by using the MPR mechanism CODES users can discuss their music and position themselves about their arguments.

e) **Proposition of the *awareness mechanisms* for understanding of the actions of other users, which provides a context for one's own actions**.
A set of mechanisms proposed and described in section 4.7, namely: a) *MPR*, (described in that section), to allow the users to get to know the reasons behind other members' actions; b) *Modification Marks*, to alert new events or changes that have happened; c) *Version Control with Layers*, to keep an explicitly (recorded and reversible) track of the steps that led to the current MP state; and d) *Action logging*, to show the history of actions (insertions, modifications, etc.) performed in the same MP. On one hand, MPR updates the users about **what** was done. On the other hand, the modification marks, version control, and action logging, allow users to understand **why** it was done. These two dimensions of awareness mechanism implemented in CODES overcome the drawbacks found in related work about providing others with the knowledge of their actions and musical ideas.

f) **Definition and construction of the CODES environment** [1] **as a proof of concept**.
CODES was designed, implemented, and evaluated in order to test more systematically the validity of this thesis' assumptions and propositions.

Some of these contributions are original, proposed in this thesis by the first time; some are well-known concepts defined by others, but their use in an integrated way in computer music is original.

The idea of not only integrate CM, CSCW or HCI concepts, but also to make experiments, was a proposal of this research in order to test and evaluate this integration itself.

The experiments and the results of the evaluation carried out in this research have shown that the CODES approach enables efficient use of the interface and

[1] temporarily hosted in the UFRGS server at the url: http://gia.inf.ufrgs.br/CODES3/.

interaction mechanism to allow lay people to experiment with music and collaborate. However, some proposals presented here, although developed for a specific musical context, can be useful to any other non-technical product collaboratively designed in an artistic domain. For example, a rationale mechanism like MPR might be useful to discuss what, and why, a scene must be present or not on a clip, if a group of video makers need to work together remotely. Also, it can be applied for the argumentation about the development of a movie storyboard or comics, collectively developed in the context of a group.

Indirect and potential contributions of this work can outline CODES as a system for social or musical inclusion and creation of virtual communities for interested in sharing musical experiences. Because group members may have different background aesthetics and skills CODES encourages direct involvement in a variety of musical experiences with music from different cultures.

6.2 Limitations

Some factors not scrutinized in this research may affect the general outcome of the MP process, and can be considered as limitations of this thesis.

One of them is related to the reliability and scalability issues. Nowadays, it is unknown wether the system behaves under stated conditions for a specified period of time as well as its capacity to handle growing amount of work.

Also, it must be stated that the lack of wide evaluation (currently, as a public tool in a unrestricted membership context) of CODES and its target public could indeed reveal new results and other requirements not observed up to the point. Specifically, aspects related to the server size, bandwidth, response time, performance, and security, for example (some of them mentioned in section 1.3), that must be investigated in further experiments are not within the scope of this research.

This research has not chosen the synchronous communication and interaction for two main reasons. On the one hand, due to the implications of implementing complex structures to control concurrent access. On the other hand, due to the nature of the musical activity itself, in which musical ideas can occur at anytime and the participants do not need to be online at the same time, since the system can store and retrieve the actions performed by the users.

Moreover, with respect to some interface issues related to computer music concepts not implemented as yet, such as tempo (capacity to setup the speed at which music is, or ought to be, played), amount and fixed duration of the sound patterns (currently as MP3 files with four seconds of duration), integration of the sound pattern editing level, sound exportation to other formats, etc. These could probably bring new perspectives to the system regarding the capacity for manipulating musical information.

6.3 Perspectives and Future Directions

Several difficulties were addressed in this work for bridging the gap between groups of novice users and the possibility of actually making music collectively over the Web. One of the main challenges comprised the mechanism for effective collaboration, allowing users to communicate with each other, discuss about their actions, and position themselves by means of *musical prototyping rationale*.

However, a set of additional identified features, out of the scope of this thesis, could improve the system even more. They could be researched in the future and included for a more useful (and complete) environment for effective use, such as:

- The use of "folksonomy" (social tagging) as a method of collaboratively allowing the users to create their own tags for categorizing themselves the content of the sound library: It is considered as an important feature to allow "collective management by people" (or by the group, in this case) of cloud tags as a way to visualize them.

- The possibility of users uploading their own musical samples, such as MP3 files, to the CODES sound library for further use in the shared musical prototypes; Is an interesting possibility to allow users to record samples of music (playing or singing) in their mobiles, for example, and then to upload it to their sound libraries.

- The implementation of a even more specific mechanism to allow users to transmit their musical idea (called preliminarily here "sound argumentation") or write comments directly in the editing area also with the purpose of writing lyrics. In this case, CODES might also be suitable for pop music groups and bands who want to make a complete prototype of their music, including the written lyrics over the displayed timeline.

- The use of the MIDI format and markup languages for music and other standard formats, to allow compatibilities with other musical software; Implementing these features would allow CODES to import and export files from/to other systems based in these formats, and to complement them with the collaborative potential shown here, not yet present in commercial music software.

- The adoption of a common or public license to manage the creative works available for members and others to build upon legally and to share. Note that all musical material produced in CODES should be free to use and share, except in the case of commercial purposes.

- The addition of information about tempo and amplitude, even considered as a musician-oriented concept, can help to improve the flexibility and customization of the musical piece since it is available transparently for novices.

CODES has shown that Web-based networked music environments can offer much more than "consumer" possibilities for novices in music. By having integrated and adequate tools, processes, and concepts in one single environment, novice users can create musical prototypes, effectively cooperate, and experience the feeling of being the creators of their own musical culture.

In the course of the CODES project over the last four years many interesting aspects emerged which have never been considered previously, revealing a great potential of this research. Such aspects, briefly presented below, have brought new perspectives and can be defined as the future directions of the CODES for:

- **A Social Network for Music Knowledge Sharing**
  A possible scenario is: " - a user wants to tap into the collective musical mind

of their friends or peers in order to discuss about musical tips, for instance, or even to know "what", "how", and "with whom" their partners are working on". Specially, in this case, where novices are not sufficiently confident to create music themselves. Thus would be possible with a kind of "social bookmarking" at the core of the CODES social network.

- **Entertainment Environment**
  Considering that more and more people spend a lot of time on the Web looking for entertainment, it is possible (for instance) to open up the CODES database to enable other companies or even schools of music to build applications for the CODES audience, such as embedded players, editors, and plugins to process, edit, and export music files. People need to have fun. CODES offers great opportunities and possibilities for people to try, enjoy, and share their collective or individual music.

- **Online and Offline mobile interactions for creating music**
  Currently, online environments like CODES do not have as yet any consonance with the offline world. On one hand, it would be very interesting for CODES members to know that their musical partners are *online* and near to them (by making use of the GPS devices of mobile phones), bringing possibilities for knowing them personally or discussing about their music projects, and so on. On the other hand, current mobile technology is powerful enough to allow the development of musical activities almost in the same way as the PC's do. In this sense, when *offline*, users can make use of mobiles for musical activities[2] in CODES, such as registering their musical ideas, editing some musical prototype or even writing comments to send later on when connected.

- **Online Environment for Music Learning**
  CODES has potential to provide the basis for developing a number of learning processes and skills to help achieve some of the aims of music learning (e.g. listening, analyzing, interpreting, improvising, composing, etc.) . Traditionally, it has often been difficult for anyone to create their own music without first developing music reading and writing abilities. By using CODES, users can directly manipulate musical samples, and share it, listening to the musical result without needing these abilities because it is conceived for novices. It provides the users with an additional way of viewing music, by allowing them to draw their entire musical sequences as easily as dragging and dropping them, and by arranging sections of music. Since it can be shared, other users (experts, for example) can provide feedback, and users could respond to that feedback and hear the results immediately. They learn by sharing and modifying others' artifacts as well as by sharing others' knowledge by means of argumentation and exchanging ideas with peers. Some previous efforts have pointed out in such a direction. By means of interactions with, and advices from more experienced users, CODES promotes music learning by beginners, positive interdependency, and encourages collaborative learning by argumentation and interaction during the development of a MP (MILETTO; PIMENTA;

[2]Two ongoing projects, a thesis and an undergraduate dissertation about Ubiquitous Music, Concepts and Motivations by Luciano Vargas Flores and Vinicius Cadore, respectively, are currently investigating this topic.

VICARI 2005).

The artistic aspects of such a cooperation proposed here may foster an individual's musical development. Thus, the use of CODES may also become an initial stimulus for further interest in the formal study of music.

Music creation by novices is ultimately about people having fun and entertainment (and maybe also learning), not about following a fixed set of rules for music composition. It is also not a matter of composing a song from the beginning to the end (such as linear music) but it is a matter of creating one's own sound sequences (non linear music).

Through the prototypical and cooperative nature of CODES, novices may thus have the opportunity to be the actors of their own musical experiences, like experienced musicians are. The partners can cooperate not only by means of explicit actions on a shared objects space and by explicit conversation, but also by interpreting the actions and, above all, the *comments* of other actors to use in their creative process.

However, CODES is not just about supporting novice people: features built for novices help everyone whose musical skills are less than a professional musician's capability. If we think musical skills are in a continuum - people do not merely know or not know music - CODES is open and accessible to all of us, from ordinary users to musicians. Therefore, if novices can learn a lot using CODES, musicians may be just like "novices" when using CODES as well, experimenting (new) ideas and exchanging opinions.

REFERENCES

ADOBE. **Flex**. Available at: <http://www.adobe.com/products/flex/> Access in: 02 apr. 2009.

BAECKER, R. M.; GRUDIN, J.; BUXTON, W.; GREENBERG, S. **Readings in Human-Computer Interaction**: toward the year 2000. [S.l.]: Morgan Kaufmann, 1995.

BARBOSA, A. Displaced Soundscapes: a survey of network systems for music and sonic art creation. **Leonardo Music Journal**, Cambridge MA, v.13, n.ISBN 0-26275392-8, p.53–59, 2003.

BARBOSA, A. Public Sound Objects: a shared environment for networked music practice on the web. **Organised Sound**, New York, NY, USA, v.10, n.3, p.233–242, 2005.

BARBOSA, A. **Computer-Supported Cooperative Work for Music Applications**. Barcelona: Prentice-Hall, Inc., 2006.

BENINI, M.; FICHEMAN, I. K.; ZUFFO, M. K.; DEUS LOPES, R. de; BATISTA, L. Editor Musical: a case of interface usability for children. In: CELDA, 2004. **Anais...** [S.l.: s.n.], 2004. p.319–326.

BORGES, M. R. S.; CAVALCANTI, M. C. R.; CAMPOS, M. L. M. Suporte por computador ao trabalho cooperativo. In: XV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 1995, Canela, RS. **Anais...** [S.l.: s.n.], 1995. (XV Jornada de Atualização em Informática).

BRYAN-KINNS, N.; HEALEY, P. G. T. Daisyphone: support for remote music collaboration. In: NIME '04: PROCEEDINGS OF THE 2004 CONFERENCE ON NEW INTERFACES FOR MUSICAL EXPRESSION, 2004, Singapore. **Anais...** National University of Singapore, 2004. p.27–30.

BRYAN-KINNS, N.; HEALEY, P. G. T.; LEACH, J.; BROOKER, A. Mutual Engagement in Collaboration. In: ABOUT-FACE WORKSHOP - CHI, 2006, Montreal. **Anais...** [S.l.: s.n.], 2006.

BRYAN-KINNS, N.; HEALEY, P. G. T.; LEE, J. Persistence in web based collaborations. In: WWW '05: SPECIAL INTEREST TRACKS AND POSTERS OF THE 14TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 2005, New York, NY, USA. **Anais...** ACM, 2005. p.910–911.

BURGE, J.; BROWN, D. C. Reasoning With Design Rationale. In: ARTIFICIAL INTELLIGENCE IN DESIGN '00, 2000. **Anais...** Kluwer Academic Publishers, 2000. p.611–629.

BURK, P. **WebDrum**. Available at: <http://www.transjam.com/webdrum/webdrum.html>. Access in: 02 mar 2009.

CAPOBIANCO, A.; CARBONELL, N. Contextual online help: elicitation of human experts' strategies. In: SYSTEMS, SOCIAL AND INTERNATIONALIZATION DESIGN ASPECTS OF HUMAN-COMPUTER INTERACTION, 2001, London. **Anais...** Lawrence Erlbaum Associates, 2001. v.2, p.824–828.

CARROLL, J. M.; ROSSON, M. B. The paradox of the active user. In: **Interfacing thought**: cognitive aspects of human-computer interaction. Rational agent: MIT Press, 1987. p.80–111.

CARSTENSEN, P. H.; SCHMIDT, K. Computer Supported Cooperative Work: new challenges to systems design. In: IN K. ITOH (ED.), HANDBOOK OF HUMAN FACTORS, 1999. **Anais...** [S.l.: s.n.], 1999. p.619–636.

CYCLING74. **Max/MSP**. Available at: <http://www.cycling74.com>. Access in: 02 mar. 2009.

DANNENBERG, R. A Brief Survey of Music representation issues, techniques and systems. **Computer Music Journal**, [S.l.], v.17, n.3, p.20–30, 1993.

DESIGN, F. **Usability First**. Available at: <http://www.usabilityfirst.com/glossary>. Access in: 02 mar. 2009.

DUCKWORTH, W. Making Music on the Web. **Leonardo Music Journal**, [S.l.], v.09, p.13–17, 1999.

ELLIS, C. A.; GIBS, S. J.; REIN, G. Groupware - Some Issues and Experiences. In: COMMUNICATIONS OF THE ACM, 1991. **Anais...** ACM, 1991. v.34, n.1, p.38–58.

FELLBAUM, C. (Ed.). **WordNet**: an electronic lexical database. 1.ed. Cambridge, MA,: The MIT Press, 1998.

FRANKLIN, J. A. Recurrent Neural Networks for Music Computation. **INFORMS J. on Computing**, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v.18, n.3, p.321–338, 2006.

FUKS, H.; GEROSA, M. A.; PIMENTEL, M. Projeto de Comunicação em Groupware: desenvolvimento, interface e utilização. In: **XXII Jornada de Atualização em Informática**. [S.l.]: Congresso da Sociedade Brasileira de Computação, 2003. v.2, p.295–338.

FUKS, H.; RAPOSO, A. B.; GEROSA, M. A.; LUCENA, C. J. P. de. Applying the 3C model to groupware development. **International Journal of Cooperative Information Systems**, [S.l.], v.14, n.2-3, p.299–328, 2005.

FUKS, H.; RAPOSO, A.; GEROSA, M. Engenharia de Groupware: desenvolvimento de aplicações colaborativas. In: XXII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2002. **Anais...** [S.l.: s.n.], 2002. p.89–128. (XXI Jornada de Atualização em Informática, v.2).

GOOGLE. **YouTube**. Available at <http://www.youtube.com/>. Access in: 02 mar. 2009.

GRUDIN, J. Computer-supported cooperative work: history and focus. **Computer**, [S.l.], v.27, n.5, p.19–26, 1994.

GUREVICH, M. JamSpace: designing a collaborative networked music space for novices. In: NIME, 2006. **Anais...** [S.l.: s.n.], 2006. p.118–123.

HARTMANN, M. I. **Estudo e implementação de arquiteturas de software para um ambiente de prototipação musical cooperativa na Web**. [S.l.]: Universidade Federal do Rio Grande do Sul, 2006.

HOLMES, T. **Electronic and Experimental Music**. [S.l.]: Routledge, 2002.

HONING, H. From Time to Time: the representation of timing and tempo. **Computer Music Journal**, Cambridge, MA, USA, v.25, n.3, p.50–61, 2001.

IAZZETTA, F.; KON, F. Internet Music: dream or (virtual) reality. In: V SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO E MÚSICA, 1998, Belo Horizonte, Brasil. **Anais...** [S.l.: s.n.], 1998. p.69–81.

ISAACS, A.; MARTIN, E. **Dicionário de Música**. Rio de Janeiro: Zahar Ed., 1985.

KOLBITSCH, J.; MAURER, H. A. The Growing Importance of e-Communities on the Web. In: WEBIST (SELECTED PAPERS), 2006. **Anais...** [S.l.: s.n.], 2006. p.19–37.

LERAY, D.; SANSONNET, J.-P. **Assisting Dialogical Agents Modeled from Novice User's Perceptions**. [S.l.]: Springer Berlin / Heidelberg, 2007. 1122-1129p. (Lecture Notes in Computer Science, v.Volume 4693).

LOY, G. **Composing with computers**: a survey of some compositional formalisms and music programming languages. Cambridge, MA, USA: MIT Press, 1989. 291–396p.

LOY, G.; ABBOTT, C. Programming languages for computer music synthesis, performance, and composition. **ACM Computing Survey**, New York, NY, USA, v.17, n.2, p.235–265, 1985.

MEDIA, F. I. **MySpace**. Available at: <http://www.myspace.com/>. Access in: 02 apr. 2009.

MILETTO, E. M.; PIMENTA, M. S.; BOUCHET, F.; SANSONNET, J.; KELLER, D. Music Creation by Novices should be both Prototypical and Cooperative - Lessons Learned from CODES. In: BRAZILIAN SYMPOSIUM ON COMPUTER MUSIC (SBMC), 12., 2009, Recife, Brazil. **Proceedings...** SBC: Porto Alegre, 2009.

MILETTO, E. M.; PIMENTA, M. S.; HOPPE, A. F.; FLORES, L. V. Who Are the Web Composers? In: ONLINE COMMUNITIES AND SOCIAL COMPUTING, 2009. **Anais...** Springer, 2009. p.381–390. (Lecture Notes in Computer Science, v.5621).

MILETTO, E. M.; PIMENTA, M. S.; VICARI, R. M. Using CODES: cooperative music prototyping and educational perspectives. In: INTERNATIONAL COMPUTER MUSIC CONFERENCE, 2005, Tampere - Finland. **Proceedings...** SuviSoft Oy Ltda, 2005. v.1, p.387–390.

MILETTO, E. M.; SANSONNET, J. P.; PIMENTA, M. S.; BOUCHET, F. Corpus-based design of a Web 2.0 Assisting Agent. In: INTERNATIONAL WORKSHOP ON WEB-ORIENTED SOFTWARE TECHNOLOGIES, 8., 2009, San Sebastian, Spain. **Anais...** [S.l.: s.n.], 2009.

MIRANDA, E. R. **Composing Music with Computers**. 1.ed. [S.l.]: Focal Press, 2001. (Music Technology Series).

MOORE, F. R. **Elements of computer music**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1990.

NIELSEN, J. **Evaluating the thinking-aloud technique for use by computer scientists**. Norwood, NJ, USA: Ablex Publishing Corp., 1992. p.69–82.

NIELSEN, J. Enhancing the explanatory power of usability heuristics. In: CHI, 1994, New York, NY, USA. **Anais...** ACM, 1994. p.152–158.

NIELSEN, J. Enhancing the explanatory power of usability heuristics. In: CHI CONFERENCE COMPANION, 1994. **Anais...** [S.l.: s.n.], 1994. p.210.

TUCKER JR., A. B. (Ed.). **Usability Engineering**. Boca Raton, FL, USA: CRC Press, Inc., 1997. 1440-1460p.

NORMAN, D. A. **The Design of Everyday Things**. [S.l.]: Basic Books, 2002.

PREECE, J.; SHARP, H.; ROGERS, Y. **Interaction Design**: beyond human computer interaction. [S.l.]: Wiley, 2007.

RAMESH, B.; DHAR, V. Supporting Systems Development by Capturing Deliberations During Requirements Engineering. **IEEE Trans. Softw. Eng.**, Piscataway, NJ, USA, v.18, n.6, p.498–510, 1992.

RAPOSO, A.; MAGALHAES, L.; RICARTE, I.; FUKS, H. Coordination of Collaborative Activities: a framework for the definition of tasks interdependencies. In: INTERNATIONAL WORKSHOP ON GROUPWARE - CRIWG, 7., 2001, Darmstadt, Germany. **Anais...** [S.l.: s.n.], 2001. p.170–179.

ROADS, C. **The Computer Music Tutorial**. [S.l.]: The MIT Press, 1996.

RODDEN, T. A survey of CSCW systems. **Interacting with Computers**, [S.l.], v.3, p.319–353, 1991.

RUBIN, J. **Handbook of Usability Testing**: how to plan, design, and conduct effective tests. [S.l.]: Wiley, 1994.

SáNCHEZ, J. A.; STRAZZULLA, D.; PAREDES, R. G. Enhancing interaction and collaboration in multimedia rooms with multilayered annotations and telepointers. In: IHC 2008: PROCEEDINGS OF THE VIII BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2008, Porto Alegre, Brazil, Brazil. **Anais...** Sociedade Brasileira de Computacao, 2008. p.117–125.

SANSONNET, J.-P. **DIVA Toolkit**. Available at: <http://www.limsi.fr/ jps/online/diva/divahome/diva.main.htm>. Access in: 02 mar. 2009.

SCHEDEL, M.; YOUNG, J. P. EDITORIAL. **Org. Sound**, New York, NY, USA, v.10, n.3, p.181–183, 2005.

EDUSP (Ed.). **Fundamentos da Composição Musical**. 3Âª Ed.ed. [S.l.]: Faber and Faber, 2008.

SELFRIDGE-FIELD, E. (Ed.). **Beyond MIDI**: the handbook of musical codes. Cambridge, MA, USA: MIT Press, 1997.

SHNEIDERMAN, B.; PLAISANT, C. **Designing the User Interface**: strategies for effective human-computer interaction (4th edition). [S.l.]: Pearson Addison Wesley, 2004.

SHUM, S. B. **Design Argumentation as Design Rationale**. The Encyclopedia of Computer Science and Technology.ed. New York: Marcel Dekker Inc, 1996. v.35, n.Supp 20.

TOFFLER, A.; TOFFLER, H. **The Revolutionary Wealth**. [S.l.]: Alfred A. Knopf, 2006. 492p.

VYGOTSKY, L. S. **Mind in Society**: the development of higher psychological processes. [S.l.]: Harvard University Press, 1980.

W3C. **Introduction to Web Accessibility**. Available at: <http://www.w3.org/WAI/intro/accessibility.php>. Access in: 02 apr. 2009.

YAHOO. **Flickr**. Available at: <http://www.flickr.com>. Access in: 02 mar. 2009.

APPENDIX A PUBLISHED WORK BY THE AUTHOR

In the context of this thesis, several papers were accepted: one paper in a Qualis B International Journal (1 PI-B), three papers in Qualis A International Conferences (3 CI-A), one paper in a Qualis B International Conference (1 CI-B), three papers in Qualis A National Conferences (3 CN-A), and one paper in a Qualis B National Conference (1 CN-B). These papers are described in more details as follows.

Papers in Peer-Reviewed Journals

- MILETTO, E. M. ; PIMENTA, M. S. ; VICARI, R. M. ; FLORES, L. V . CODES: a Web-based Environment for Cooperative Music Prototyping. Organised Sound (Print), Cambridge University Press, v. 10, n. 3, p. 243-253, 2005.

- PIMENTA, M. S. ; MILETTO, E. M. ; HOPPE, A. F.; FLORES, L. V. ; Cooperation in Musical Prototypes Design . (Invited paper to appear in a special issue for the International Journal of Future Generation Computer Systems)

Papers in Peer-Reviewed Conferences

- PIMENTA, M.; MILETTO, E.; SANSONNET, J.P.; BOUCHET, F.; Making Social and Shared Music on the Web with CODES Web Technologies - a Track of the 25th Annual ACM SAC - IN: Symposium on Applied Computing, 22 - 26 March 2010, Sierre, Switzerland.

- SANSONNET, J.P.; MILETTO, E.; BOUCHET, F.; PIMENTA, M.; Exploring the integration of teaching capabilities into a CSCP framework through help agents IN: 20th Brazillian Symposium on Informatics in Education (SBIE), 17 - 20 nov 2009, Florianópolis, Brasil.

- MILETTO, E.; PIMENTA, M.; BOUCHET, F.; SANSONNET, J.P.; KELLER, D.; Music Creation by Novices should be both Prototypical and Cooperative - Lessons Learned from CODES. In Proceedings of SBCM 2009 - 12th Brazilian Symposium on Computer Music. Recife, Brazil, 2009.

- MILETTO, E.; SANSONNET, J.P.; PIMENTA, M.; BOUCHET, F.; Corpus-based design of a Web 2.0 Assisting Agent. In Proceedings of IWWOST 2009, 8th International Workshop on Web-Oriented Software Technologies, San Sebastian, Spain, 2009.

- HOPPE, A. F.; MILETTO, E. M. ; PIMENTA, M. S. ; FLORES, L. V. ; Cooperation in Musical Prototypes Design . In Proceedings of 13th International Conference on Computer Supported Cooperative Work in Design. IEEE Santiago, Chile, 2009.

- MILETTO, E. M. ; PIMENTA, M. S. ; HOPPE, A. F.; FLORES, L. V. ; Who are the Web composers. In Proceedings of HCI International. Springer LNCS. San Diego, 2009. (to appear)

- MILETTO, E. M. ; FLORES, L. V. ; PIMENTA, M. S. ; SANTAGADA, L. . Interfaces for Music Activities and Interfaces for Musicians are not the same: the Case for CODES a Web-based Environment for Colaborative Music Prototyping. In: The Ninth International Conference on Multimodal Interfaces (ICMI 2007), 2007, Nagoya. Proceedings do IX ICMI, 2007. p. 201-207

- MILETTO, E. M. ; FLORES, L. V . ; KUCK, D. ; PIMENTA, M. S. ; RUTILY, J . Beyond Open Source Music Software: Extending Open Source Philosophy to the Music with CODES . In: Linux Audio Conference - LAC, 2007, Berlin, 2007

- FLORES, L. V. ; MILETTO, E. M. ; KUCK, D. ; RUTILY, J. ; PIMENTA, M. S. . Interfaces Musicais não são Interfaces para Músicos: Discussão e Projeto de uma Interface Musical para Leigos. In: 11 Simpósio Brasileiro de Computação Musical, 2007, São Paulo. Anais do XI Simpósio Brasileiro de Computação Musical, 2007.

- MILETTO, E. M. ; FLORES, L. V. ; RUTILY, J. ; PIMENTA, M. S. CODES: Supporting Awareness in a Web-based Environment for Collective Music Prototyping. In: Simpósio de Fatores Humanos em Sistemas Computacionais, 2006, Natal. Anais do IHC2006, 2006

- MILETTO, E. M. ; PIMENTA, M. S. ; FLORES, L. V. ; VICARI, R. M . CODES: Um Ambiente para Prototipação Musical Cooperativa Baseado na Web. In: XXXII Seminário Integrado de Software e Hardware, 2005, São Leopoldo. Anais do XXV Congresso da Sociedade Brasileira de Computação, 2005.

- MILETTO, E. M. ; PIMENTA, M. S. ; COSTALONGA, L. L. ; VICARI, R. M. Construindo Protótipos Musicais Cooperativamente na Web. In: 10 Simpósio Brasileiro de Computação Musical, 2005, Belo Horizonte. Anais do 10 Simpósio Brasileiro de Computação Musical. Belo Horizonte : PUC-MG, 2005. p. 82-93.

- MILETTO, E. M. ; PIMENTA, M. S. ; VICARI, R. M. Using CODES: Cooperative Music Prototyping and Educational Perspectives. In: International Computer Music Conference, 2005, Barcelona. Proceedings of ICMC 2005. Tampere - Finland : SuviSoft Oy Ltda, 2005. v. 1. p. 387-390

Other Related Publications

- COSTALONGA, L. L. ; VICARI, R. M. ; MILETTO, E. M. . Agent-Based Guitar Performance Simulation. Journal of the Brazilian Computer Society, v. 14, p. 19-30, 2008.

- COSTALONGA, L. L. ; MIRANDA, E. R. ; MILETTO, E. M. . Octopus Music API: Modelling Musical Performance. In: 11 Simpósio Brasileiro de Computação Musical, 2007, São Paulo. Anais do XI Simpósio Brasileiro de Computação Musical, 2007.

- PIMENTA, M. S. ; MILETTO, E. M. ; Dietrich, C. ; NEDEL, L. P. . O Theremin Virtual: Usando Dispositivos de Realidade Virtual em Experimentos Musicais. In: Décimo primeiro Simpósio Brasileiro de Computação Musical, 2007, São Paulo. Anais do XI Simpósio de Computação Musical, 2007.

- COSTALONGA, L. L. ; MILETTO, E. M. ; FLORES, L. V. ; VICARI, R. M. Bibliotecas Java Aplicadas a Computação Musical. In: 10 Simpósio Brasileiro de Computação Musical, 2005, Belo Horizonte. Anais do $10\frac{1}{4}$ Simpósio Brasileiro de Computação Musical. Belo Horizonte : PUC-MG, 2005. p. 70-81.

- COSTALONGA, L. L. ; FLORES, L. V. ; MILETTO, E. M. ; VICARI, R. M. Simulação de Performances de Violão por Agentes Artificiais. In: 10 Simpósio Brasileiro de Computação Musical, 2005, Belo Horizonte. Anais do 10 Simpósio Brasileiro de Computação Musical. Belo Horizonte : PUC-MG, 2005. p. 374-379.

APPENDIX B FORM OF THE QUALITATIVE TEST

This appendix presents the questions posed in the qualitative test that were filled out after the CODES usage.

6.4 Open Questions - Cooperation using Versioning Tree

1. Is the look and feel of CODES pleasant?
2. Do you keep informed about what is happening?
3. Are the expressions and the language used in the interface clear and easy to understand?
4. Did you get lost at some moment or something happened that you did not expected?
5. Is the feedback adequately presented and easy to interpret?
6. Does the graphical sound representation in CODES help to identify the sound content even without listening to it?
7. Do you think this sound representation (icons) could be done in any other way?
8. Would you like to associate your own icons to the sound patterns?

APPENDIX C CODES INTERFACE EVOLUTION

This appendix contains some excerpts of the CODES design process as described in the section 3.1.

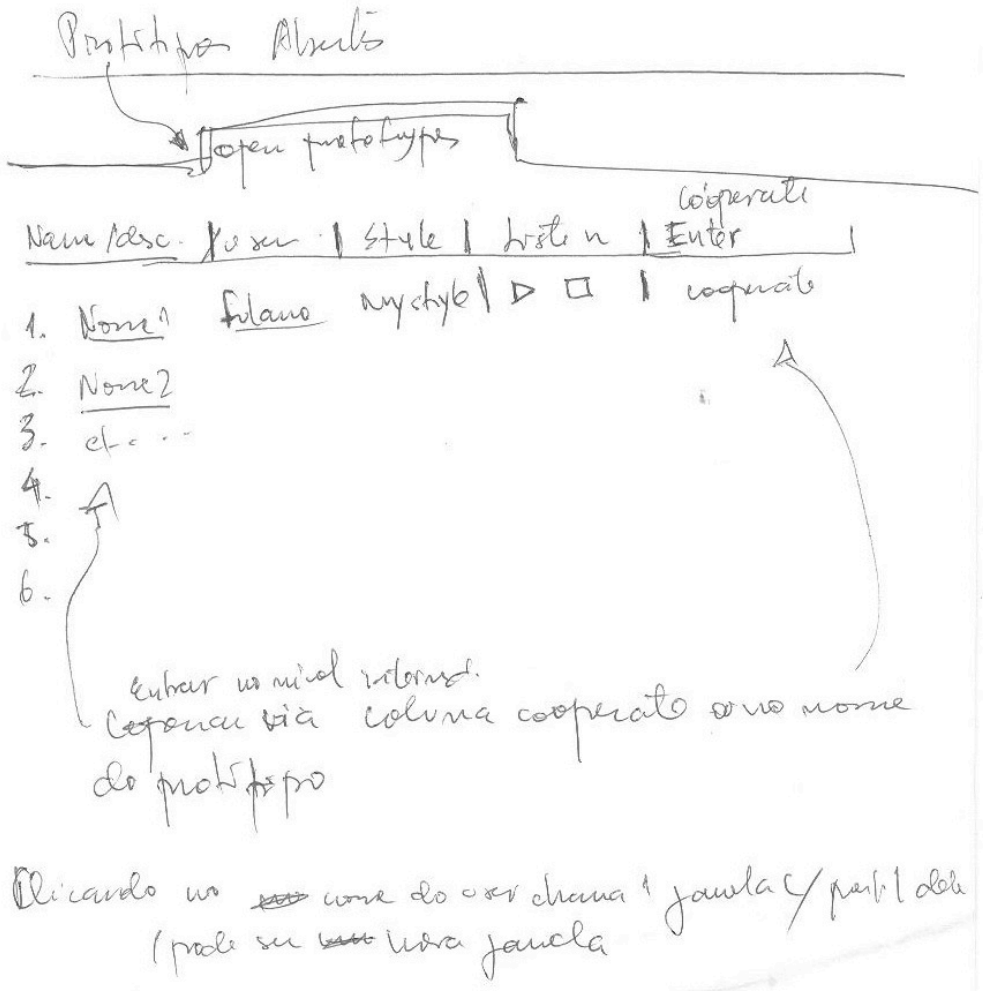


Figure 6.1: First sketch of the CODES Public Level

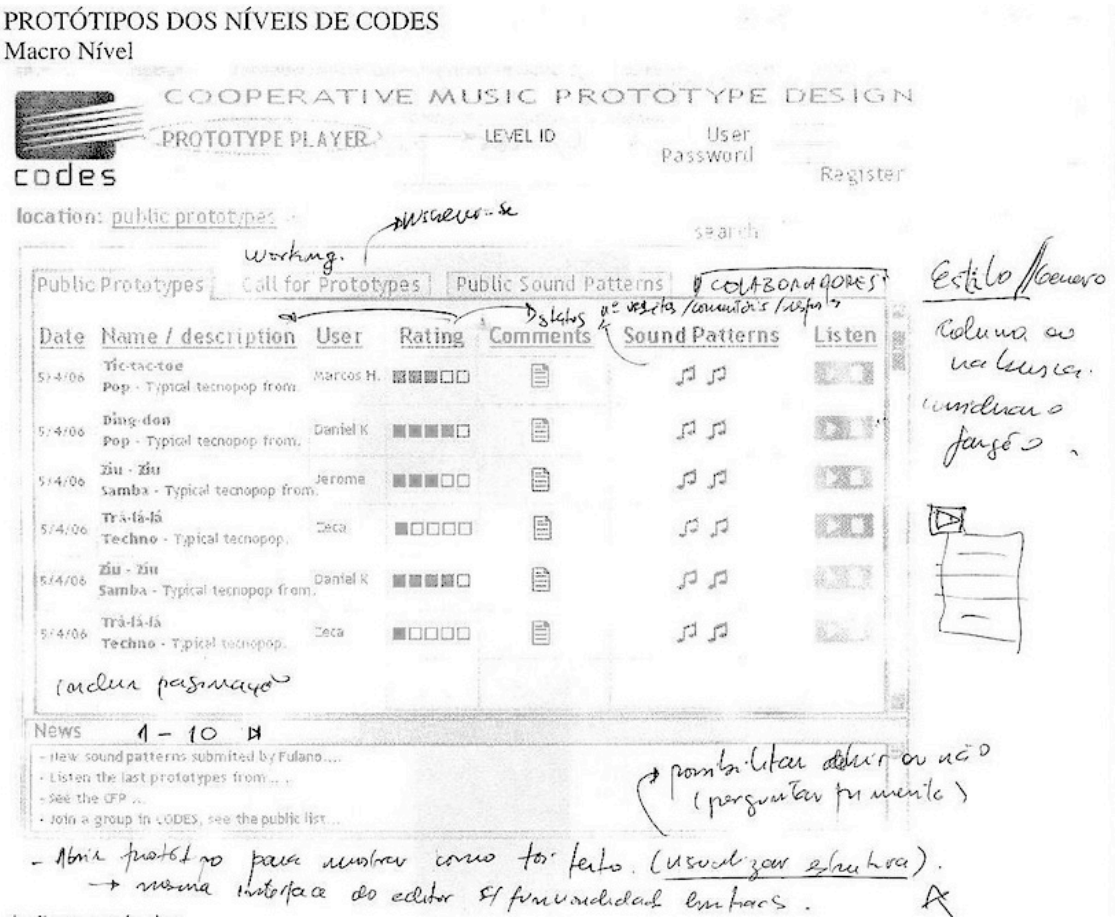


Figure 6.2: Last sketch of the CODES Editing Level

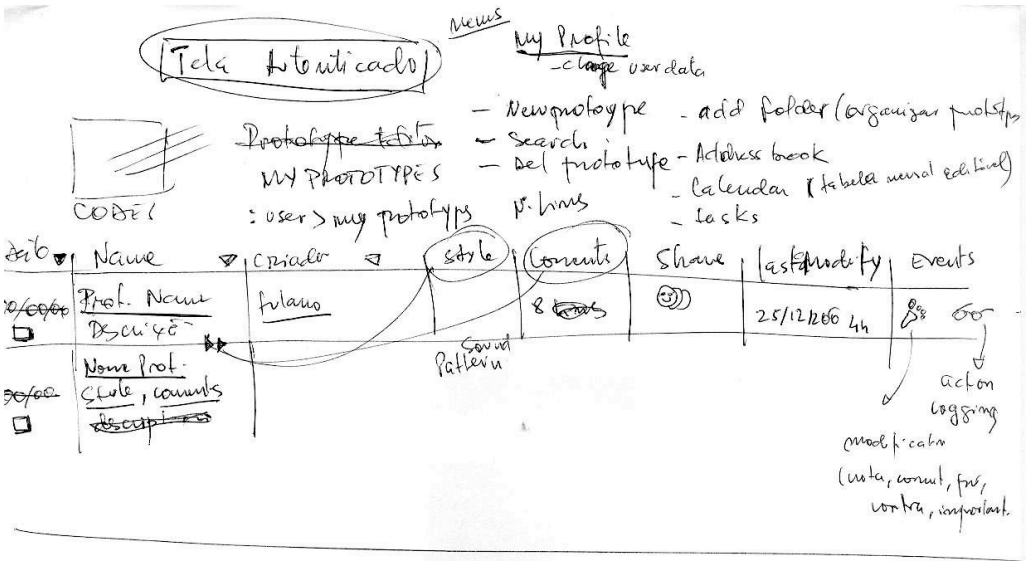


Figure 6.3: Second sketch of the CODES Public Level

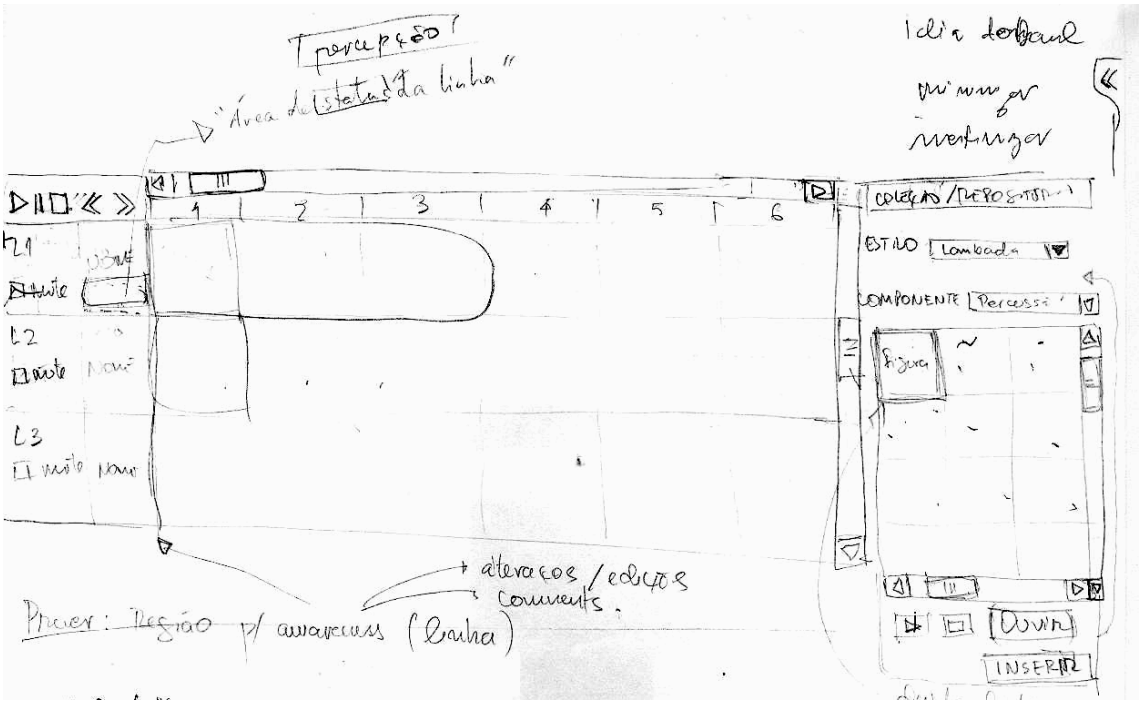


Figure 6.4: First sketch of the CODES Editing Level

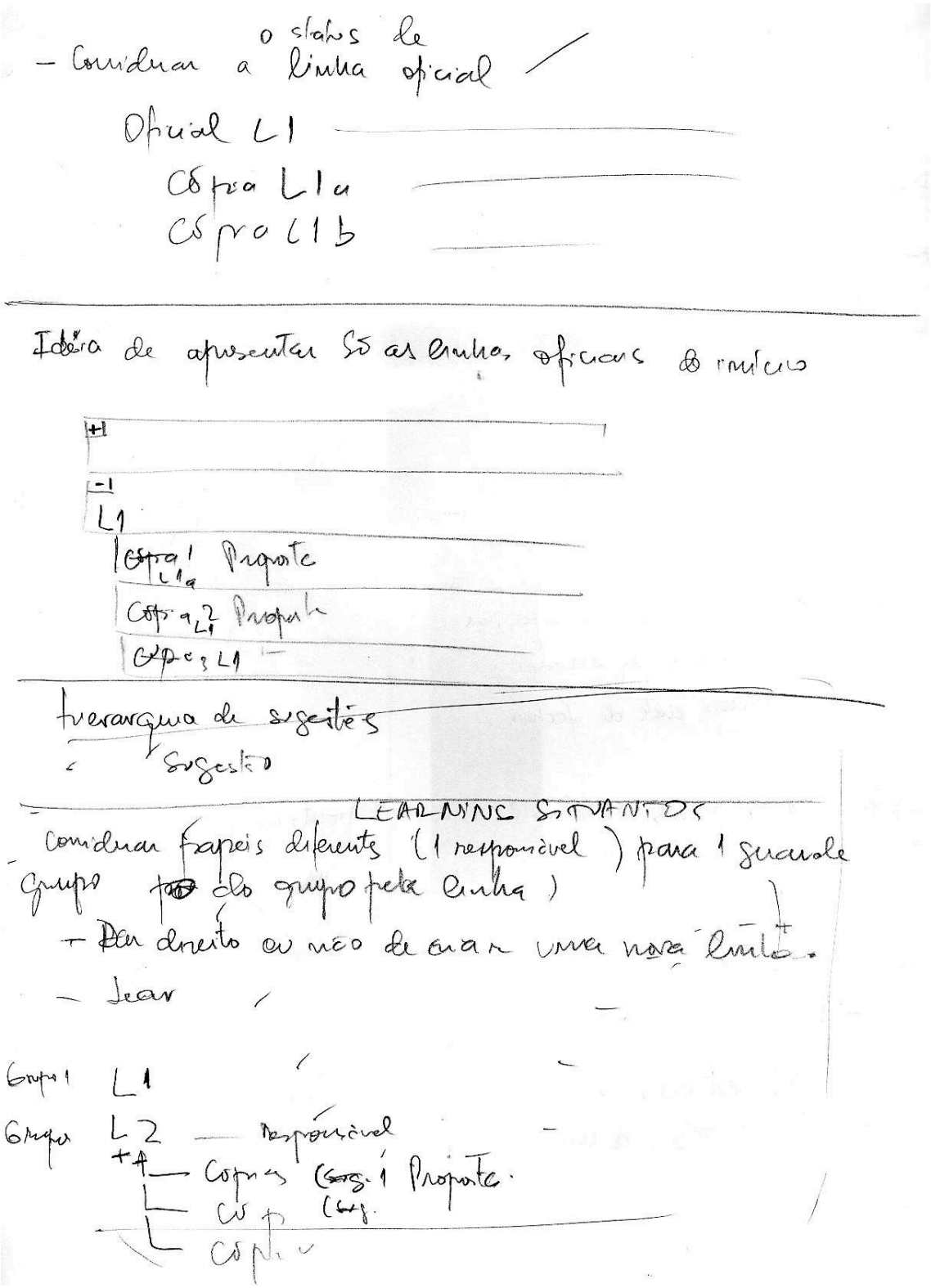


Figure 6.5: Sketch of the early CODES Editing Level with tracks

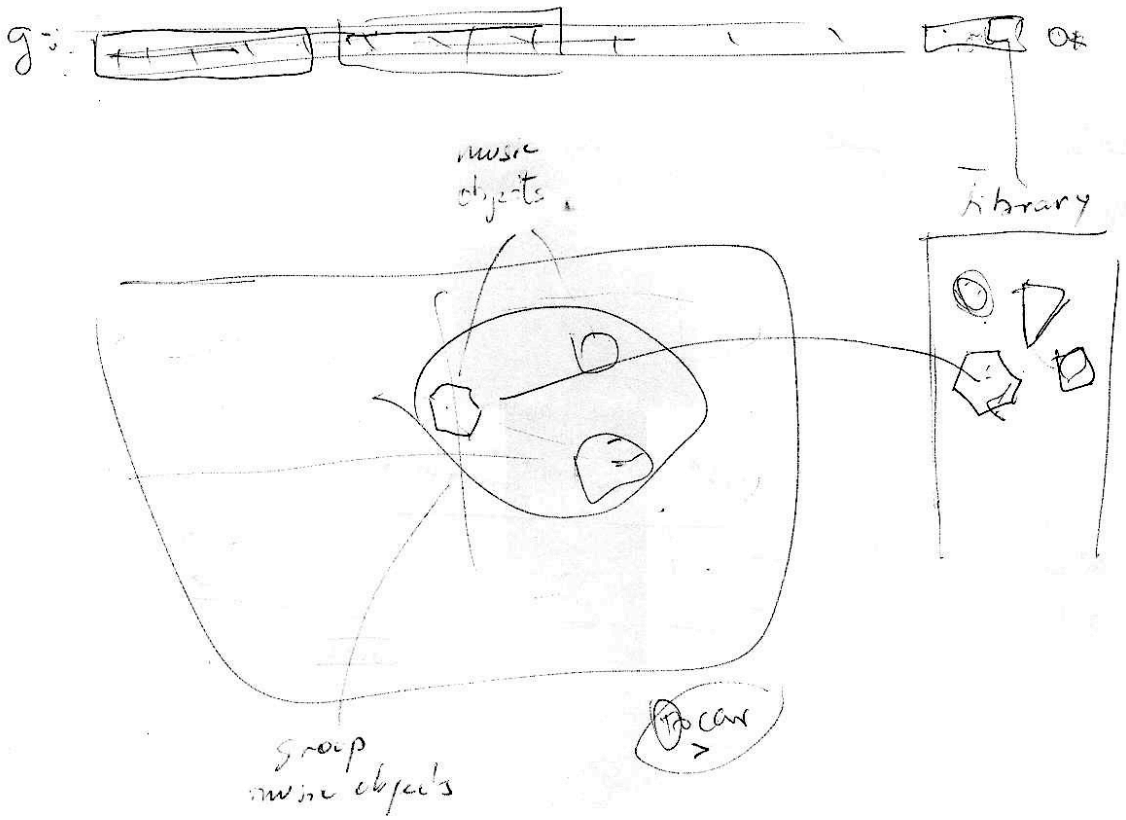


Figure 6.6: Sketch of user interaction in CODES Editing Level (drag and drop)

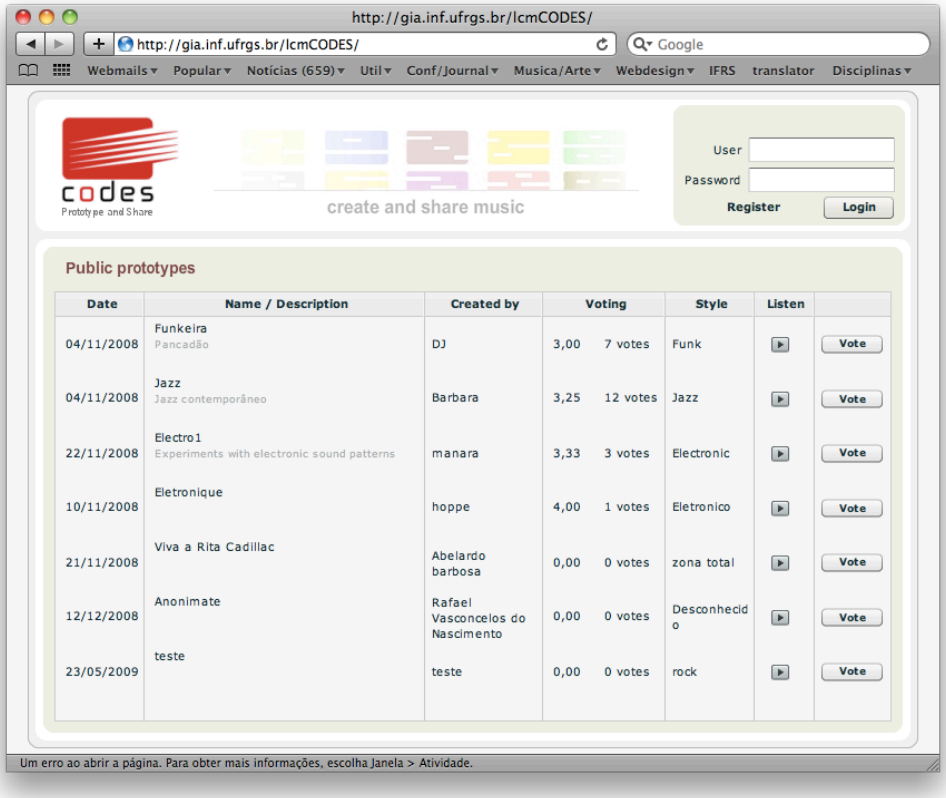


Figure 6.7: Old version of CODES public Level (JavaScript and Flash)

APPENDIX D CODES SCENARIOS

This appendix presents the main scenarios related to CODES usage. They are presented according to the CODES main activities, namely *exploring, creating, editing, inviting, sharing*, and *publishing*.

6.5 Exploring activities in the CODES home page

6.5.1 Scenario: Browse published music

Actor: User.
Normal Flow

1. *User* enters the query in the search form field choosing one criterion: "prototype", "musical style", and "author"; click in search
   System validates the query; retrieves the information; displays it in the home page.

1'. *User* scrolls down the page; click in the "next page", "previous page" hyperlink;
System (updates the new page) displays the data in the home page.

Alternate Flow

1". *User* enters a invalid query in the search form field;
System displays the search results ("0" in this case).

6.5.2 Scenario: Listen to published music

Actor: User.
Normal Flow

1. *User* clicks the play button of the selected music;
   System starts playing the music;

2. *User* clicks the stop button while playing the selected music;
   System stops playing the music;

6.5.3 Scenario: Rate published music

Actor: User.
Normal Flow

1. *User* moves the mouse pointer over the "stars" (representing the values to be rated) of each music prototype;
   System displays the labels of the correspondent values ("Poor", "Nothing special", "Worth", "Pretty cool", "Awesome";)

2. *User* chooses and clicks on a "star" to rate the music;
   System displays the label; updates the value of the general rate;

6.5.4 Scenario: Register in CODES

Actor: User.
Normal Flow

1. *User* clicks on "Register" hyperlink;
   System opens the register window; ask the user data; ask to choose a identification color;

2. *User* fill out the form; choose the color; click register;
   System validates the data; displays a confirmation of the action to the user;

Alternate Flow

2'. *User* enters a invalid data in a mandatory fields; choose a color already in use;
   System validates the data; displays the feedback to the user informing the fields to be corrected; displays a confirmation of the action to the user;

2". *User* cancels the operation clicking in the close button;
   System validates the data; displays the feedback to the user informing the fields to be corrected; displays a confirmation of the action to the user;

6.5.5 Making a tour in CODES

The "Tour in CODES" is a section accessible at the CODES home page (hyperlink *Tour in CODES* on the top of the page), whose goal is to give a system overview.

1. *User* clicks on the hyperlink *Tour in CODES* in the CODES home page.
   System opens the *Tour in CODES* windows;

2. *User* clicks on the tabs (hyperlinks) to know the system functionalities and characteristics;
   System changes the content, according to the selected tab;

6.5.6 User Authentication in CODES

Actor: User.
Normal Flow

1. *User* types login and password
   System authenticates; shows a screen with the user prototypes list;

Alternate Flow

1'. *User* enters an invalid data;
   System validates the data; displays the feedback to the user, informing the incorrect fields;

6.6 Exploring the Intermediate Level

In CODES, the intermediate level is a screen presented after the user login, whose the goal is to allow to the users to manage their musical prototypes, checking new contributions, and new actions performed by others partners as well.

6.6.1 User edits its user profile

Actor: User.
Normal Flow

1. *User* clicks on the hyperlink "edit user profile"
   System opens the corresponding form with the fields (name, e-mail, login, password, color id, musical preferences);

2. *User* fill out the form; confirms clicking on the Save button;
   System confirms; gets back to the intermediate level;

Alternate Flow

2'. *User* leaves blank fields or enter invalid data (login, color);
   System validates the data; displays the feedback to the user, informing the correct procedure;

6.6.2 User creates a new musical prototype

Actor: User.
Normal Flow

1. *User* clicks on the hyperlink "create new musical prototype"
   System opens the corresponding form with the fields (name, style, and description) ;

2. *User* fill out the form; confirms clicking on the corresponding button;
   System authenticates; opens the editing area with the new musical prototype created;

Alternate Flow

2'. *User* leaves blank fields;
   System validates the data; displays the feedback to the user, informing the correct procedure;

6.6.3 User checks the contributions

Actor: User.
Normal Flow

1. *User* select one of their musical prototypes'
   System shows the corresponding versions of the selected musical prototype ; shows a feedback informing to the available options (prototype properties, invite user, delete prototype)

2. *User* select one version of their musical prototypes'
   System shows the corresponding contributions of the selected version ; shows a feedback informing the user the available options (listen to, delete prototype, add new version, invite user, publish version)

3. *User* select one or several contribution of a selected version
   System shows the information (properties); shows a feedback informing to the available options (listen to, edit, delete prototype, contribution properties, add new version, invite user, publish version)

6.6.4 User deletes a musical prototype

Actor: User.
Normal Flow

1. *User* select a musical prototype
   System shows the corresponding versions of the selected musical prototype ; shows a feedback informing the user the available options (prototype properties, invite user, delete prototype)

2. *User* clicks on "delete" a musical prototype
   System validate the action; asks for confirmation;

3. *User* confirms the action
   System displays a feedback message; excludes the musical prototype;

Alternate Flow

2'. *User* clicks on "delete" other's musical prototype;
   System validates the action; checks if it is the owner; displays the feedback to the user, informing that is not possible; comeback to the musical prototypes list;

6.6.5 User listens to contributions

Actor: User.
Normal Flow

1. *User* selects a musical prototype
   System shows the corresponding versions of the selected musical prototype; shows a feedback informing the user the available options (prototype properties, invite user, delete prototype)

2. *User* selects one version of their musical prototypes
   System shows the corresponding contributions of the selected version ; enables the "Listen" option that will play the whole prototype.

3. *User* clicks on "Listen" button
   System starts to play the musical prototype;

4. *User* selects one or several contributions of the selected version
   System enables the "Listen" option that will play the selected contributions.

6.6.6 User invites partners to contribute

Actor: User.
Normal Flow

1. *User* triggers the "Invite user" option
   System shows the invitation screen and corresponding form fields;

2. *User* fill out the form; choose the option (to search CODES members or invite external users by mail); confirms the action
   System checks the data; completes the action; sends a feedback to the user.

6.7 Exploring the Music Prototype Editing Level

In CODES, the music prototype editing level allows user to access and manipulate the CODES sound library, the editing and cooperative functionalities.

6.7.1 User manipulates the sound library

Actor: User.
Normal Flow

1. *User* selects a musical style
   System shows the corresponding sound patterns;

2. *User* selects a sound pattern (clicking on it)
   System plays the corresponding audio;

6.7.2 User uploads sound file (not implemented)

Actor: User.
Normal Flow

1. *User* selects the "upload" function at the music prototyping editing level;
   System shows the respective form ;

2. *User* fill out the form choosing a local sound file; user clicks "upload" button;
   System checks the file type; checks the file size; set the file in the musical style chosen by the user;

Alternate Flow

2'. *User* chooses an invalid file;
   System validates the action; sends a feedback indicating the problem; prompt the form to redo the action;

6.7.3 User edits a musical prototyping

Actor: User.
Normal Flow

1. *User* drags and drops a sound pattern from the sound library to the editing area;
   System puts the respective icon in the editing area;

2. *User* resizes the horizontal length of the icon;
   System sets the size of the icon and the respective sound file;

3. *User* replaces the position of the icon;
   System sets the new position in the editing area; adjusts the time to start to play the file;

4. *User* deletes a sound pattern;
   System excludes the sound pattern of the editing area; excludes the sound file of the play list;

6.7.4 User saves the musical prototype

Actor: User.
Normal Flow

1. *User* chooses the "save" option;
   System shows the corresponding form;

2. *User* fills out the form; chooses one of the saving options (draft, shared, or new version); clicks in the "save" button;
   System validates the options; saves the file; sends a feedback to the user;

6.7.5 User invites partners to contribute

Actor: User.
Normal Flow

1. *User* triggers the "Invite user" option
   System shows the invitation screen and corresponding form fields;

2. *User* fill out the form; choose the option (to search CODES members or invite external users by mail); confirms the action
   System checks the data; completes the action; sends a feedback to the user.

6.7.6 User checks the log entries

6.8 Exploring the Sound Pattern Editing Level

In CODES, the sound pattern editing level allows user to create and edit the sound patterns. This functionality was partially implemented in CODES.

6.8.1 User creates and edit a sound pattern

Actor: User.
Normal Flow

1. *User* chooses the "create sound pattern" option;
   System shows the corresponding screen; enables the functionalities;

2. *User* writes the sound with the mouse pointer;
   System set the sounds to be played;

3. *User* chooses a midi instrument to play the sound; *System* set the instrument;

4. *User* chooses the “play” option;
   System plays the corresponding sound;

5. *User* selects an area with the mouse pointer, which he wants change something; user chooses the “delete” option; *System* marks the area; excludes the representing sounds;

6. *User* saves the sound pattern; attributes it to a style in the sound library;
   System validates the action; puts the sound pattern in the CODES sound library;

APPENDIX E RESUMO ESTENDIDO

Um Ambiente Cooperativo e Interativo na Web para Prototipação Musical

6.9 Introdução

Computação Musical é uma disciplina da Ciência da Computação que inclui teoria e aplicação de tecnologias (novas ou existentes) na música. Isso envolve subáreas como acústica, síntese sonora, projeto sonoro, processamento de sinais digitais, difusão sônica, psicoacústica, composição mediada por computador, entre outras. Em consequência, computação musical é uma área interdisciplinar com grande potencial investigativo em outras áreas (MOORE 1990).

Cooperação na Internet é um desses tópicos que atualmente abrem muitas possibilidades investigativas. O que antes era considerado como ferramenta auxiliar apenas para composição musical por especialistas (IAZZETTA; KON 1998), hoje a Internet transforma e influencia definitivamente a música contemporânea, despertando a atenção também de usuários leigos em música.

Essa tendência de cooperação cresce integrada com o campo da computação musical num novo domínio, que foi recentemente chamado de *networked music* (música em rede) - uma convergência potencial de atividades sociais e do "fazer-musical" - tema de uma edição especial do *Organised Sound Journal* (2005).

Por um lado, as aplicações de *networked music* descritas em uma *survey* por Barbosa (2003), associadas com a Web 2.0 permitem a artistas experimentais (como leigos em música) explorar as implicações da interconexão de computadores para fins musicais. Usuários passivos passam a ser ativos produtores de conteúdos e atores da experiência do usuário final. Por outro lado, novos requisitos específicos são necessários para que ocorra uma efetiva cooperação e experimentação musical por estes leigos.

No contexto desta tese, alguns passos são dados em direção a melhora da experiência do usuário, indo além da "produção" e "compartilhamento" de arquivos. Um ambiente baseado na Web chamado CODES (*COoperative Musical Prototypes DESign*) foi projetado e desenvolvido para permitir aos leigos experimentação musical cooperativa visando a possibilidade de torná-los "criadores de conteúdo musical".

Os experimentos realizados com o CODES revelaram vários desses requisitos que surgiram durante os testes, demonstrando uma necessidade de uma abordagem ampla de IHC, não apenas para tornar o design de interação do CODES mais eficiente, mas também para proporcionar um método de trabalho comum para área de *networked music*.

Assim, esta tese em ciência da computação sugere a integração das subáreas de computação musical, interação humano-computador e trabalho cooperativo suportado por computador para permitir cooperação efetiva em experimentos musicais via Web por leigos em música.

6.9.1 Motivação e Objetivos

Existem barreiras naturais a serem eliminadas para envolver apropriadamente esses usuários leigos em experimentos com ações musicais cooperativas. A lista a seguir enumera algumas dessas barreiras e os respectivos desafios desta pesquisa. Elas são:

a. **Criação musical**: como proporcionar criação musical por leigos em música?

b. **Interfaces inadequadas**: como proporcionar a prática de ações do domínio da tecnologia musical para leigos?

c. **(In)acessibilidade universal**: como proporcionar acesso universal a usuários dispersos considerando requisitos e limitações tecnológicas?

d. **Notação musical**: como apresentar informações musicais de fácil entendimento e num alto nível de forma que possam ser usadas por leigos em música?

e. **Possuir um instrumento musical**: como usar e dominar um instrumento musical virtual para produzir o som desejado?

f. **Dificuldade na manipulação musical**: como descobrir, encontrar, combinar e sequenciar amostras musicais, editar sequências, armazenar e recuperar esses sons?

g. **Perda de contextualização**: como representar, distinguir e identificar novas ações e contribuições dos outros no contexto de um grupo? como proporcionar aos outros o entendimento das suas idéias musicais e ações?

h. **Falha na comunicação**: como relacionar argumentos e decisões de usuários com decisões e ações por eles realizadas de forma a permitir o refinamento e evolução de um produtor musical interativo?

i. **Falta de negociação**: como proporcionar negociação efetiva entre usuários para atingirem um resultado final consensual?

Para resolver esses problemas um "processo de prototipação musical" é proposto nesta tese e um sistema Web denominado CODES que foi desenvolvido e testado com usuários reais.

CODES foi concebido e projetado para dar suporte à prototipação musical cooperativa (CMP) com enfoque especial ao usuário leigo em música, possibilitando-lhe a realização de experimentos musicais individuais ou coletivos. Aspectos relacionados à interface, interação e cooperação são propostos, apresentados e discutidos ao longo do texto, visando preencher várias lacunas encontradas nos trabalhos relacionados.

O objetivo principal desta tese é investigar como a tecnologia que envolve interações musicais em rede pode proporcionar um suporte adequado para criação musical e discutir como é possível eliminar um conjunto de barreiras naturais que

impedem essas interações, bem como definir requisitos especificamente orientados a usuários leigos em música.

Como resultado principal e prova de teste desta pesquisa, CODES é introduzido para dar suporte à criação musical cooperativa por leigos em música. Para tanto, este trabalho propõe novos conceitos e características com os quais a tecnologia pode oferecer grandes contribuições para formas sociais da realização musical por leigos em música. Como principal motivação desta pesquisa está a crença de que não é necessário conhecimento musical prévio a qualquer leigo interessado em criar protótipos musicais.

6.9.2 Contribuições

Principais contribuições desta tese:

- Definição e aplicação dos princípios de prototipação e de cooperação para estimular as atividades de criação musical orientadas a leigos em música;
- Integração e adaptação de conceitos, métodos e técnicas de computação musical, IHC e CSCW;
- Definição de requisitos de interface para leigos interessados em experimentos musicais;
- Proposta do conceito de Prototipação Musical Cooperativa como processo de iterativo e interativo de experimentação musical coletiva;
- Proposta de um mecanismo de argumentação denominado *music prototyping rationale* para dar suporte à exposição de idéias e motivos que guiam as ações musicais dos usuários dentro do ambiente CODES.
- Proposta de um mecanismo de versões visando fornecer negociação efetiva e manter a autoria de cada contribuição do usuário;
- Definição e construção do ambiente CODES como prova de conceito.

6.9.3 Escopo

Esta tese está focada em aspectos tecnológicos realizados durante os quatro anos de pesquisa.

As propostas e contribuições aqui apresentadas focam mecanismos, tecnologias e conceitos relacionados a arquiteturas, interfaces e aspectos interativos das áreas de computação musical, IHC e CSCW.

Desta forma, os seguintes aspectos estão fora do escopo desta tese:

- Estética e qualidade musical.
- Aspectos de escalabilidade do ambiente CODES.

6.10 Fundamentos e Trabalhos Relacionados

Esta seção apresenta um resumo dos conceitos e tecnologias relacionados ao tema principal da tese. A coleção de trabalhos relacionados, analizada, resumida e descrita na Tabela 2.1 retrata o atual estado das principais aplicações musicais em rede na Web orientadas a leigos música.

Não há, entretanto, uma classificação estabelecida para esta comparação, até onde se sabe, devido a heterogeneidade dos sistemas, mesmo os considerando como parte do universo de aplicativos baseados na Web para colaboração musical.

A comparação é apresentada de acordo com critérios associados a atividades de cooperação musical na Web para leigos e fundamentos escritos no capitulo 2.

A lista de trabalhos relacionados inclui os sitemas JamSpace (GUREVICH 2006), Pitch Web (DUCKWORTH 1999), EduMusical (BENINI et al. 2004), WebDrum (BURK 2009), Daisyphone (BRYAN-KINNS; HEALEY 2004) e PSO (BARBOSA 2005). Embora todos estes sistemas sejam concebidos para permitir interações musicais cooperativas, a comparação na Tabela 2.1 mostra lacunas importantes, particularmente:

- os problemas típicos relacionados com a colaboração efetiva, tais como argumentação, autoria de contribuições, awareness, memória de grupo e persistência não são adequadamente considerados e explorados;

- há uma necessidade de integração de conceitos de outras áreas para que sistemas úteis e usáveis com foco no público leigo sejam projetados no domínio musical, tais como usabilidade, acessibilidade, design de interação, *awareness* e *design rationale*, entre outros.

- sistemas não estão aptos a suportar argumentação e negociação para usuários colaborarem na criação musical;

- sistemas não oferecem mecanismos adequados de *awareness* ou trilha de interação visando fornecer feedback adequado para que o grupo entenda as contribuições dos usuários;

- sistemas não são adequadamente projetados para suportar a transmissão de idéias musicais de um usuário para um grupo;

- sistemas atuais não tratam ou desconsideram autoria das contribuições, o que não permite aos usuários retomarem suas contribuições originais em qualquer momento;

Considerando todas essas lacunas, o principal problema nesse campo parece estar na falta de foco na cooperação efetiva bem como numa interface projetada para leigos. Ainda, há uma falta de conhecimento sobre requisitos, reais necessidades e tarefas de usuários leigos em música para interações musicais cooperativas. As características de CODES e o processo de prototipação musical cooperativa aqui proposto se constituem em esforços para completar essas lacunas e resolver esses problemas.

6.11 CODES: uma Visão Geral

Este capítulo apresenta e discute os requisitos para sistemas direcionados a usuários leigos para atividades musicais bem como a arquitetura Web adequada para suportar criacção coletiva de conteúdo e interação de grupo.

A filosofia geral do projeto de CODES é começar com uma plataforma tecnológica específica (Web) e área de aplicação (protótipos musicais), e em seguida, alavancar suas potencialidades para encontrar as possíveis interações que atendam as exigências do cenário prototipação musical cooperativa.

A abordagem de CODES para a cooperação entre os usuários, a fim de criar MPs coletivos e para permitir a argumentação e discussão, permite a cada usuário entender os princípios e as regras envolvidas no complexo processo de criação e experimentação musical.

Através da tecnologia de computador, CODES fornece uma forma eficaz de quebrar algumas barreiras para iniciantes que desejam envolver-se em prototipação e experimentação musical. O ponto aqui é a mera possibilidade de "criar" e não da qualidade final do trabalho.

Além disso, em CODES também foi investigado o uso da Função de Assistência (particularmente ACA), uma solução amplamente conhecida e utilizada por alunos em situações de aprendizagem. No entanto, esta função aplicada no contexto de ajudar usuários leigos nas tarefas musical não é algo comum.

No projeto CODES, o desafio tem sido o de fornecer apoio as atividades musicais sem preocupação com a realidade do músico, porque leigos claramente não agem e/ou pensam da mesma forma que os músicos. Além disso, as metáforas e conceitos usualmente adotados em ambientes baseados em computador para representar elementos de atividades musicais (notas, melodia, harmonia, ritmo, timbre, etc.) em geral não são bem compreendidos por leigos em música. Isto pode ser visto como um bom ponto de partida para a definição de um processo específico orientado a leigos para criar música cooperativa.

6.12 Prototipação Musical Cooperativa usando CODES

Este capítulo apresenta o suporte de CODES para o processo de prototipação musical cooperativa na Web e uma descrição das questções que esta atividade compreende. CODES fornece suporte para a cooperação na Web para atividades não-técnicas realizadas por iniciantes, o que implica uma abordagem diferente para essa cooperação.

Por um lado, leigos não têm conhecimento suficiente e confiança para criar música. Sob tais circunstâncias, eles precisam experimentar com base na tentativa e erro, que é a essência da prototipação e do design. Por outro lado, uma atividade não-técnica tal como a música criada em CODES necessita apoiar a negociação não-sistemática e oportunista a fim de ajudar os leigos e envolvê-los em suas interações no processo de prototipação musical.

Com o apoio fornecido por CODES, os usuários podem obter uma melhor compreensão das atividades de criação musical através da abstração de conceitos e representação musical alternativa, bem como através da interação com outros usuários, experientes ou não, a fim de obter feedback.

A motivação para prototipação musical cooperativa vem do desafio de permitir

que leigos criem música por si mesmos. Esta atividade pode ser realizada integrando-se conceitos de computação musical, IHC e CSCW. Isso tem permitido a implementação de uma solução que inclua os aspectos cooperativos e prototípicos no processo de criação de música de CODES, uma vez que permite a manipulação da representção musical da mesma forma como em um processo comum de prototipação.

A expectativa é de que a cooperação através dos mecanismos de CODES pode oferecer possibilidades interessantes para os parceiros incentivando o interesse dos usuários na música, até mesmo a aqueles auto-considerados iniciantes.

6.13 Experimentos e Avaliação

Os testes do ambiente CODES envolveram diferentes temas e objetivos a fim de se obter um feedback geral sobres aspectos relativos a usabilidade, acessibilidade e cooperação.

Os resultados são promissores.

No primeiro experimento, testes realizados visaram descobrir problemas de interação e interface, bem como sobre a necessidade de manter a autoria da contribuição original de cada usuário. A Figura 6.8 ilustra a soma dos índices de pesquisa de opinião feitas sobre o sistema, através do questionário apresentado na seção 5.1.5.

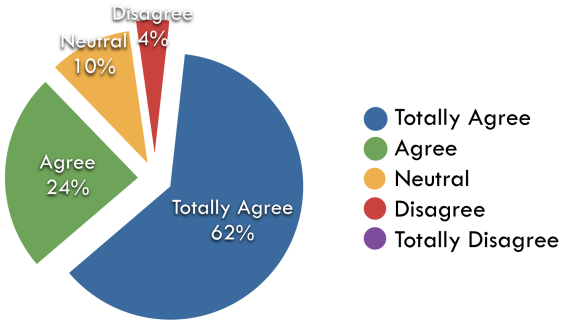


Figure 6.8: Índice de aprovação geral do ambiente CODES

As legendas da figura, listadas da melhor para a pior hipótese, representam as seguintes alternativas: *totally agree* = concordo totalmente, *agree* = concordo, *neutral* = neutro, *disagree* = discordo, *totally disagree* = discordo totalmente.

CODES foi projetado para ser o mais simples possível para leigos envolvidos em experimentos musicais cooperativos na Web. Considerando que a maioria das senteças foram afirmativas durante os experimentos, esta pesquisa considerou que o objetivo foi atingido com sucesso.

É necessário conhecer o público-alvo e suas necessidades (leigo em música), a fim de projetar CODES para fornecer o suporte adequado. Baseado nesta idéia e após estes experimentos, novos requisitos orientados a leigos, descritos na seção 4.4, foram apresentados como parte do projeto CODES.

Analisando os trabalhos relacionados, uma das principais vantagens de CODES está no suporte à cooperação efetiva para estes leigos.

Os trabalhos relacionados são ou se consideram como "sistemas musicais em rede" (*networked music systems*) basicamente por razões técnicas (tais como fácil

acesso, publicação de material, suporte de servidor, manutenção, etc.); enquanto esta pesquisa considera outros aspectos, além destas razões.

Em sistemas cooperativos como CODES situado no domínio das artes, considerado como um domínio não técnico, o uso de *design rationale* é imperativo uma vez que os usuários não conhecem o resultado final dos seus trabalhos. Esta pesquisa propõe uma adaptação deste conceito, aqui definido como *Music Prototyping Rationale*, associando-o com requisitos específicos orientados a leigos para interações musicais.

CODES basea-se no foco do processo - o qual é adequado para leigos - e não na estética do resultado final.

Os experimentos foram planejados para serem executados em contexto restrito, entretanto, dados extra surgiram no corpo dos resultados gerais. Mesmo sabendo e conhecendo estes termos e condições, alguns usuários sentiram a necessidade de envolver e convidar novos parceiros, motivados por si próprios, após conhecer e experimentar CODES. Isto, permite inferir que a) o propósito geral da proptotipação musical cooperativa foi bem aceito, uma vez que leigos querem criar sua própria música, independente de seu conhecimento prévio; b) mesmo apresentando ainda algumas dificuldades iniciais, devido a curva de aprendizado, o suporte de CODES à proptotipação musical cooperativa é bom o suficiente para estimular usuários a convidar seus amigos como parceiros ou membros de sua música e motivá-los a usar CODES para melhorar e compartilhar seus experimentos musicais e c) de forma geral, as pessoas tem necessidade de encontrar novas formas de expressão. CODES é apenas mais uma (boa) maneira de permitir isso.

6.14 Conclusão e Trabalhos Futuros

O desenvolvimento do projeto CODES permitiu eliminar os obstáculos apresentados na seção introdutória (6.9) deste capítulo e da tese (Capítulo 1), da seguinte maneira:

a. desenvolvendo uma interface com usuário orientada a leigos, integrando conceitos de IHC e CSCW para possibilitar atividades de prototipação e cooperação; assim, a barreira da interface de tecnologia musical para especialista foi eliminada;

b. usando a internet e recomendações sobre arquiteturas padrão para fornecer acesso universal;

c. usando representações musicais alternativas para quebrar a barreira do conhecimento da notação musical;

d. criando um ambiente multinínvel para possibilitar o uso de PC como um instrumento virtual e eliminar a barreira de possuir e portar um instrumento musical tradicional;

e. propondo um processo de manipulação direta para sistematizar a experimentação musical, tanto individual quanto coletiva;

f. adaptando conceitos de IHC e CSCW como “awareness” e “design rationale”, para propor mecanismos específicos para leigos realizarem interação e cooperação efetiva uns com os outros, quebrando assim a barreira da visualização, argumentação e entendimento das ações musicais;

g. criando um mecanismo de gerência de versões a fim de manter as contribuições originais dos autores.

Como resultado, essa tese proporcionou uma série de contribuições resumidas na próxima seção.

6.15 Sumário de contribuições

Todas as contribuições desta tese tiveram como objetivo a resolução dos problemas e lacunas descritos na seção 1.1 (motivação e objetivos) em comparação com os trabalhos relacionados;

A contribuição geral desta pesquisa reside num novo ponto de vista de como deve ser a infraestrutura tecnológica para fornecer suporte a criação musical para leigos. Para tanto, considerando os aspectos cooperativos e prototípicos aqui descritos, o ambiente CODES foi desenvolvido como prova de conceito para fornecer suporte e permitir experimentação musical relacionando estes aspectos e outras idéias relacionadas. CODES incorpora uma abordagem para permitir uma cooperação efetiva entre usuários leigos que necessitam suporte adequado para produzirem e discutirem sua música.

As investigações no decorrer deste trabalho têm produzido contribuições relevantes, a saber:

a) **A definição e aplicação de dois princípios - prototípico e cooperativo - para envolver e estimular atividades criativo-musicais direcionados a leigos**;

b) **Integração e adaptação de conceitos, métodos e técnicas de computação musical, IHC e CSCW**;

c) **Definição de características e requisitos de sistema para experimentação musical por leigos**;

d) **Proposta do conceito de "Prototipação Musical Cooperativa" como um processo incremental e interativo para criação de protótipos musicais**;

e) **Proposta de mecanismos de awareness para entendimento das ações dos outros usuários, fornecendo um contexto para suas próprias ações**. Um conjunto de mecanismos propostos e descritos na seção 4.7, apresentam: a) *MPR*, (music prototyping rationale), para permitir aos usuários o conhecimento sobre as razões por trás das ações dos outros membros; b) *Marcas de Modificação*, para alertar novos eventos ou alterações ocorridas; c) *Controle de Versão com Camadas*, para manter um explícito (gravado e reversível) histórico de passos que guiaram o estado atual do protótipo musical; e d) *Log de ações*, para mostrar o histórico de ações (inserções e modificações) realizadas no mesmo protótipo musical.

f) **Definição e construção do ambiente CODES como prova de conceito**.

Algumas contribuições são originais, propostas pela primeira vez nesta pesquisa; outras são conceitos definidos por outros, porém com o seu uso integrado de forma original na área de computação musical.

A idéia de não apenas integrar conceitos, mas também de realizar experimentos foi uma proposta desta pesquisa com a finalidade de testar e avaliar esta integração. Os experimentos e os resultados da avaliação desenvolvidos nesta pesquisa mostraram que a abordagem usada em CODES possibilita o uso eficiente dos mecanismos de interação e interface para permitir que leigos possam realizar experimentos musicais colaborativos.

Embora algumas propostas apresentadas nesta pesquisa tenham sido desenvolvidas num contexto musical específico, elas podem ser usadas em outros contextos envolvendo produtos não técnicos colaborativamente projetados para um domínio das artes.

6.16 Limitações

Alguns fatores não examinados nesta pesquisa podem afetar o resultado geral do processo de prototipação musical e serem considerados como limitações desta tese.

Um deles está relacionado aos problemas de escalabilidade e confiabilidade. Atualmente, não se sabe o comportamento do sistema em relação a sua capacidade de lidar com uma crescente carga ou volume de processamento.

Ainda, deve-se registrar que a falta de uma avaliação mais ampla impede que se conheça novos resultados e novos requisitos não observados até o presente momento. Principalmente aspectos relativos a capacidade do servidor, largura de banda, tempo de resposta, performance e segurança por exemplo que devem ser investigados em futuros experimentos e não estão no escopo desta tese.

Ainda, com respeito a aspectos de interface relacionados a conceitos de computação musical não implementados tais como tempo, quantidade e duração fixa dos padrões sonoros, integração do nível de edição de padrões sonoros, exportação de som para outros formatos, etc., estes provavelmente trariam novas perspectivas para o sistema tendo em vista a capacidade de manipulação musical.

6.17 Perspectivas e Trabalhos futuros

Várias dificuldades foram abordados neste trabalho visando minimizar ou eliminar os problemas entre grupos de usuários novatos e a possibilidade de realmente fazer música em grupo através da web.

Um desses principais desafios inclui o mecanismo de colaboração efetiva, permitindo a comunicação entre usuários, discussão sobre suas ações e seu posicionamento por meio de *musical prototyping rationale*.

Todavia, um conjunto de caractersísticas adicionais, fora do escopo desta tese, poderiam melhorar o sistema ainda mais. Entre elas, destacam-se:

- o uso da folcsonomia (social tagging) como método para permitir colaborativamente que usuários criem suas próprias tags e categorias diretamente na biblioteca de sons;
- a possibilidade de usuários fazerem upload de seus próprios arquivos de sons para a biblioteca CODES;
- a implementação do mecanismo específico para transmissão de idéias musicais chamado preliminarmente de *sound argumentation*;

- o uso do formato MIDI e outras linguagens de marcação padrão para música visando compatibilidade com outros software;

- a escolha de uma licença pública para gerenciar trabalhos criativos para membros e outros desenvolverem e compartilharem seus experimentos amparados por questções legais.

- a funcionalidade do tempo e amplitude, mesmo considerada como um conceito orientado a músico, pode ajudar e melhorar a flexibilidade e customização da peça musical se disponível para leigos de forma transparente;

Os aspectos artísticos dessa cooperação aqui propostas podem favorecer o desenvolvimento musical de um indivíduo. Assim, o uso de CODES também pode se tornar um estímulo inicial para desenvolver o interesse no estudo formal de música.

A criação musical por leigos está relacionada, em última instância, a pessoas tendo diversão e entretenimento (e talvez também aprendizagem) e não em seguir um conjunto fixo de regras para a composição da música. Também não é uma questão de compor uma música do início ao fim (de forma linear), mas é uma questão de criar suas próprias seqüências de som (música não linear).

Através da natureza prototípica e cooperativa de CODES, leigos podem ter a oportunidade de ser os atores de suas próprias experiências musicais, como são músicos experientes. Os parceiros podem colaborar não apenas por meio de ações explícitas em um espaço de objetos compartilhados ou pela conversa explícita, mas também pela interpretação das ações e, acima de tudo, dos comentários de outros atores para usar no seu processo criativo.

Contudo, CODES não trata apenas sobre o apoio a usuários leigos: características construídas para ajudar todos os novatos cujas habilidades musicais são inferiores a capacidade de um músico profissional. Ao pensar em habilidades musicais num contínuo, CODES está aberto e acessível a todos nós, de usuários comuns a músicos. Portanto, se os leigos podem aprender usando CODES, os músicos podem ser como "leigos" quando utilizarem CODES, experimentando (novos) intercâmbios de idéias e opiniões.