CODES: a Web-based environment for cooperative music prototyping

EVANDRO MANARA MILETTO, MARCELO SOARES PIMENTA, ROSA MARIA VICARI and LUCIANO VARGAS FLORES

Organised Sound / Volume 10 / Issue 03 / December 2005, pp 243 - 253
DOI: 10.1017/S1355771805000981, Published online: 29 November 2005

Link to this article: http://journals.cambridge.org/abstract_S1355771805000981

How to cite this article:
doi:10.1017/S1355771805000981

Request Permissions : Click here
This paper presents CODES – COoperative Music Prototype DESign, a Web-based environment for cooperative music prototyping. Its main goal is to allow any user – especially those with no expertise in music – to draft musical pieces collectively, in a prototyping manner. So, such musical sketches – we call them music prototypes – can be repeatedly tested, listened to, and modified, not only by their original creators but also by the online partners that will cooperate in their refinement, until their final form is reached. CODES enables sharing of knowledge by means of rich interaction and argumentation mechanisms associated to each prototype modification, which are also secure ways of providing awareness to this asynchronous collaborative environment. In this paper, we present the concept of music prototyping and introduce the main aspects related to cooperative prototyping of musical pieces, focusing on issues concerning a musical piece as a collective creation of a virtual community. We will also show some usage examples as a means to describe the overall architecture, behaviour and potentials of the CODES environment.

1. INTRODUCTION

Music technology has undergone considerable changes over the last decade, mainly with the increasing use of the Internet. Even presenting some constraints as for sound information traffic, the Internet is becoming increasingly attractive as a tool for music making (Kon and Iazzetta 1998). An example of its possibilities is networked music, as described in a survey by Barbosa (2003), which allows experimental artists to explore the implications of interconnecting their computers.

Day by day the Internet is growing as an environment for communication, data exchange, and information in all fields, shortening distances and providing interaction facilities that support the growth of virtual communities. Indeed, Internet-based networked music has gained wider acceptance, and the existing applications have evolved towards more sophisticated projects and concepts including, for example, real-time performance systems and different systems for multi-user interaction and collaboration.

In this paper we introduce CODES – COoperative Music Prototype Design, a Web-based environment for cooperative music prototyping. CODES aims at allowing users to make music experiments and interact with each other in order to create simple musical pieces (herein named music prototypes or, simply, prototypes – see section 2).

The CODES project associates concepts of computer music, human-computer interaction (HCI), and computer supported cooperative work (CSCW) to allow lay people interested in music to experience and reinterpret the sense of creating and developing their own musical culture and skills through the Web. Like Weinberg (2002), we are interested in providing any user (from non-music experts and children to experienced musicians) an access to meaningful and engaging musical experiences. Through CODES, non-music experts may have the opportunity to be – like experienced musicians are – the actors of their own musical experiences. This means they can create musical examples (prototypes) that can be tested, modified, and repeatedly listened to, both by the first authors and by their partners that will be cooperating in the refinement of the prototype. Moreover, such a cooperation may foster the individual’s musical development, and so using CODES may also become an initial stimulus to further interest in the formal study of music.

This paper is organised as follows. The music prototyping concept is introduced in section 2. Section 3 presents some reported experiences with cooperative musical composition. The CODES system, its architecture, and main characteristics are presented in section 4. Some characteristics of the CODES user interface and interaction are presented in section 5. Section 6 highlights the support for cooperative activities. Different uses for CODES are considered in section 7, and a final discussion, in section 8, concludes the paper.

2. MUSIC PROTOTYPING

In industry, prototyping is usually adopted as a cyclic process for the creation of a simplified version of a product, so that its characteristics and processes of conception and construction can be understood. Prototyping aims at creating successive versions of
the same product incrementally, providing new improvements at each new version.

However, in the musical field, some peculiarities make the creation and conception process different from those carried out in other fields. Musical composition is a complex activity where there is no agreement about what activities have to be done and in which sequence: each person has a unique style and way of working, and then most composers have not yet developed the tradition of sharing their musical ideas and collaborating while composing.

‘Prototyping’ is not a common expression in music literature. In fact, the activity carried out by composers is usually called ‘composition’. But, in principle, non-specialists in music are not composers and the results of their creative experiments are deliberately called ‘music prototypes’ in this paper in order to highlight the difference. In the music literature, ‘draft’ is commonly applied to such a kind of creative work, but our emphasis is mainly on the process (prototyping), and not on the product itself (in which case ‘prototype’ or ‘draft’ correspond to the same idea).

In our opinion, music is an artistic product that can be designed through prototyping. A musical idea (a note, a set of chords, a rhythm, a structure or a rest) is created by someone (typically for a musical instrument) and afterwards cyclically and successively modified and refined according to his/her initial intention or to ideas that come up during the prototyping process. Besides musicians, non-specialists in music are also probably interested in creating and participating in musical experiments, but they lack environments oriented to their profile. In fact, we believe no previous musical knowledge should be required by any user to create music prototypes.

3. RELATED WORK

This section summarises the characteristics of some environments found in the literature for collective musical composition. Clearly, most of these environments are for music composition by music specialists, rather than for music prototyping by non-specialists. The major motivation underlying our proposal is to allow non-specialists in music to access a virtual space in order to interact with each other, explore sounds together, discuss this exploration, and retrieve all the discussed information any time they want.

In a survey about IMNs – interconnected musical networks – Weinberg proposes four different levels of interconnectivity among participants, based on the roles the computer plays in enhancing their independent social relations: ‘The Server’, ‘The Bridge’, ‘The Shaper’, and ‘The Construction Kit’ (Weinberg 2002). Most of the Internet-based systems for music composition described here are cited in that survey within the last level, ‘The Construction Kit’. In this level, there is high interconnectivity among participants, who are usually experienced musicians. Participants are allowed to provide their own material and manipulate (listening, altering, refining, etc.) others’ contributions, usually in an asynchronous interaction and offline material manipulation.

The PIWeCS (Whalley 2004) system is a complex composition system based on a dialogue between human and non-human agency. PIWeCS integrates intelligent agents with Max/MSP software through a Web user interface. The system would have to work in a physical space like an art gallery, but also online for off-site participants, and be suited to combine both points of access, whilst CODES was designed to run in a virtual space only via a Web browser to eliminate the barrier of geographic distance among partners (physical presence).

The FMOL system (Jordà 2000) is related to real-time collaborative musical composition on the Web. Using a plug-in, the system allows many distributed users to work together in one or more musical works. Collaboration is carried out with a vertical multi-track model. FMOL implements interesting and complex synthesis concepts that are more indicated for skilled users instead of concepts that were designed to be useful and usable for non-music experts as in CODES.

The EduMusical system (Ficheman 2002) 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 between students in virtual classrooms, guided by a tutor. In contrast, CODES allows independence from a tutor and non-structured groups, despite the possibility of supporting structured groups with a tutor role usually necessary in learning situations (Miletto, Pimenta and Costalonga 2004).

The TransJam system (Burk 2000) aims at allowing musicians, who are connected to its website, to make musical performances together, selecting loops of the instruments that are to be played. Despite the focus being on supporting performance, there is some simple support for composition here. Daisystone (Bryan-Kinns 2004) is an environment for remote group music improvisation presenting a novel design for more engaging social and serendipitous musical environments. Daisystone focuses on the representation of looping music, and provides support for remote collaboration, and support for the formulation of ideas. Both TransJam and Daisystone are systems based on a looping metaphor that could present some disadvantage related to style and rhythm flexibility. CODES tries to overcome this by offering users the possibility of creating multiple lines with different music styles and rhythms, and even mixing them in the same prototype.
PitchWeb (Duckworth 2000) is a multi-user musical instrument specifically designed for the Internet. Basically, it uses plane geometric shapes (figures) that can be selected and manipulated, and finally mapped onto sound samples that need to be downloaded to the browser cache before starting the system. The user interaction in the environment allows for choosing the order in which samples will be played, represented by the shapes. There are other systems that deal with composition, like Creating Music (Subotnick 2004) and HyperScore (Farbod, Pasztor and Jennings 2004), which enable non-musicians to compose (or create), collectively or not. Our approach differs from these works mainly regarding HCI issues. They need additional plug-ins and software installed (like Shockwave Flash and QuickTime Player) to run. CODES is designed to be accessible to a great diversity of platforms and browsers (all W3C compliant – see W3C 2004), minimising requirements of use and thus increasing accessibility.

Rolf Wöhrmann and Guillaume Ballet’s (1999) article examines client-server architectures for computer music, drawing upon the example of IRCAM’s Studio Online project. The authors outline a number of issues in distributing sound-processing and database services over the World Wide Web. The system they describe is addressed to skilled musicians instead of non-music experts as in our case. Our approach differs from these works mainly regarding HCI issues. They need additional plug-ins and software installed (like Shockwave Flash and QuickTime Player) to run. CODES is designed to be accessible to a great diversity of platforms and browsers (all W3C compliant – see W3C 2004), minimising requirements of use and thus increasing accessibility.

Some aspects – mainly regarding technological concerns – are common to most of these systems:

- adoption of a client-server architecture;
- use of the MIDI sound format;
- implementation in a platform-independent Java language – except FMOL, implemented in C++ and available only for Windows or Linux operating systems; and
- unrestricted access to ordinary users, i.e. any user may login or access the system without an additional fee.

Most of these systems could be used by non-experienced users; in practice, however, more skilled users may obtain better results if they have specific knowledge about sound synthesis (in the cases of FMOL and the system described by Wöhrmann and Ballet) or if the activities are carried out under a tutor’s supervision (in the case of EduMusical).

In addition to common characteristics of other systems (such as the ones briefly discussed above), CODES addresses three other aspects that are very important to consider in a collaborative environment for music composition/prototyping:

1. **Group awareness:** mechanisms to manage understanding of actions and decisions of group members. In fact, similar works do not have any support for group awareness. More details about the CODES group awareness mechanisms are presented in section 6.

2. **Support to long prototyping sessions:** an important mechanism in prototyping sessions – in fact, in any design activity – is the capacity of interrupting a session and resuming it in order to continue the process from the last break point. A music prototyping session can take many days or even weeks before a final result is reached.

3. **Export/import sound format:** providing sound representation formats in order to allow export/import musical pieces/prototypes from/to one environment/system to/from others. Currently available formats are the well-known MIDI and Wave. Standard MIDI was chosen here due to easy manipulation and compatibility. Although the sounds of synthesised MIDI files played on most PCs are still low quality, it yields some future possibilities, like conversion from MIDI to common music notation. We are investigating the use of some markup languages for music – like MusicXML, Music Markup Language (MML), Music Encoding Initiative (MEI), and Standard Music Description Language (SMDL) – as interesting alternatives to be explored. We believe that in the near future one of them (or some variation thereof) will be the standardised format of choice for music content on the Web.

One of the crucial aspects that gives CODES an advantage over the systems described above is the communication concept that is necessary to implement cooperative work, which we discuss in section 6.

### 4. THE CODES ENVIRONMENT

CODES is an environment for cooperative musical prototyping on the Web, designed to be used by people interested in music who wish to prototype and share their musical ideas. A major requirement of CODES is that it should support music prototyping in such a cooperative way that users would not need to be skilled musicians or to have any specialised knowledge of music to use CODES and create music prototypes. In this section we will present a brief description of its architecture and implementation aspects.

CODES is based on the classical client-server architecture (figure 1). On the client side, the Sonic Manipulation Manager Applet manipulates sound (sound files selection, mixing, playing, stopping, etc.), sending user events to CODES managers on the server side. Actions such as invitations, adding comments to musical pieces, and event perception are manipulated by the Cooperation Manager, together with the User Manager, to verify user’s authentication. Sonic patterns are organised on the server side through the Sonic Pattern Manager, which reads the directories,
updates and retrieves files that are required by the Sonic Manipulation Manager. All activities related to the user (login, register, authentication) are performed by the User Manager, which communicates with the Cooperation Manager to execute cooperation actions among users and groups. The Database Manager provides access to the database, where MIDI files, application data (messages, logs, etc.), and user data are stored.

CODES implementation follows a free software philosophy, aiming at providing easy access to software development tools. The Java Sound API, from Java programming language, allowed us to focus system development on both graphical user interface (GUI) and cooperation aspects, making the sound handling easier because of the components that already offer sound control. The Web server used was TomCat Servlet Container (TomCat 2005), a reference for Java applications that accesses a MySQL (MySQL 2005) database through a JDBC connection (JDBC 2005).

5. CODES USER INTERFACE AND INTERACTION

The CODES user interface was designed to cover aspects related to interaction flexibility, robustness, and ease of use, as well as to present adequate support when complex musical information is displayed, thus providing an effective interaction between users and environment. The environment was designed to reach a balance between user interfaces that are so ‘easy’ for the user that they end up depleting their expressiveness, and others that are so complicated that they discourage beginners (D’Arcangelo 2002).

CODES understands a musical prototype as formed by lines (tracks) of instruments and arrangements, such as bass, arpeggios, drum lines, etc. Editing is typically made by selecting sonic patterns among pre-defined patterns available in CODES. A user is allowed to create more than one line, which means that someone can be the ‘owner’ of more than one line (like user 1 in figure 2). By clicking the Play button, the Sonic Manipulation Manager starts the execution of the selected lines, that is, all lines enabled for playback (option Mute unselected – see muted line in figure 2). All chosen patterns vertically grouped in the same timeline are mixed and played, under complete user control (see time 3 in figure 2), which can stop and restart at any time with the usual control buttons (Play, Stop, Forward, Rewind, Pause).

User interaction, therefore, basically includes actions such as ‘selecting’ (by clicking) and playing sonic patterns, and combining them with other patterns selected by the ‘partners’ (other users) of the same music prototype. This combination can occur in different ways: overlapping (simultaneous playing), juxtaposition (sequencing), etc. Sonic patterns are high-level musical structures (small parts of musical files in MIDI format) that make the process of sound choice and prototyping easier. The possible patterns of choice for a cell have small sonic differences, but keep the same style and same duration, which makes it easier for the user to adapt them to his/her piece.

Many musical elements are pre-defined in the CODES system patterns, including concepts such as rhythm, tempo, melody, harmony and timbre. Users do not need to know the conventional music notation (score) to create prototypes: they may select, play and combine such patterns in an interactive way by direct manipulation.
Other user interface details can be identified in figure 2. Three users (user 1, user 2, and user 3) are the track owners of the four existing lines L1, L2, L3, and L4. The Post-it icon (see line comments in the figure) indicates the presence of a user note or of an argumentation related to some action carried out in that line. Lines L3 and L4 belong to other users, and a possible action for the active user (user 1) is to mute them during execution, by selecting the check box with the label Mute, as in line L4.

Users may begin building their musical prototype in CODES by choosing a new line, which will represent the music genre or style. Such choice will determine what kind of sound pattern will be made available for the users of that particular music prototype (for example: pop rock, bossa nova, blues, jazz, etc.). An overview of a line creation can be seen in figure 3. The first step when creating a new line is to open the ‘Edit’ menu and click ‘Add line’ (figure 4). A new window will open and the user will choose a musical style for that line (see the example of a jazz style in figure 5). Then, the user has to choose what we refer to as a ‘musical component’, where the options are rhythm, bass, notes, and base (chord) (figure 6). Each cell created in the line will launch the same collection of sound patterns, according to the users’ choices. Thus, they can combine the sequence they consider the most adequate.

6. COOPERATIVE ACTIVITIES IN CODES

Cooperative music prototyping is here defined as an activity that involves people working together on a musical prototype. Cooperation in CODES is asynchronous, since it is not necessary to manage the complexity of real-time events for the development of musical prototypes. Users can access the prototype, do their experiments, and write comments at different times and from any location, via a Web browser.

CODES considers that a musical prototype is initiated by someone, the prototype owner. He or she uses CODES to elaborate an initial musical prototype and to ask for the collaboration of other ‘partners’, by sending explicit invitations (typically using e-mail features). Partners who accept the invitation can participate in the collaborative musical refinement and manipulation.

The coordination (one of the characteristics of cooperative work) of all activities in such a musical context can happen naturally, when the group recognises one member as having more musical abilities, being more experienced or simply willing to take the
lead. We believe that it is not necessary to make distinctive and explicit representation of coordinator role, because hierarchisation of group actions and communications is not our intention. Usually, more experienced users’ opinions and actions in a group with an explicit coordinator role may inhibit the other users’ participation.

The group of partners of a prototype in CODES may evolve into a virtual community, and so CODES may be considered as a communityware (Liechti 2000) for entertainment. Typically, communityware aims at supporting the formation of informal groups of people and at supporting the interactions in such communities. In order to provide support for these interactions, CODES offers three kinds of awareness mechanisms:

1. Music prototyping rationale: to allow users to link their explanations with their actions on music prototypes;
2. Action logging: to maintain an explicitly recorded trail of the steps that led to the current prototype state; and
3. Modification marks: to indicate to a user that a prototype has been modified by others.

In fact, awareness mechanisms offer several benefits to music prototyping:

- keeping track of decisions;
- tracking progress in music prototyping and identifying conflicts, which may initiate a negotiation process among multiple points of view;
supporting the building of cumulative prototyping knowledge;
- assisting the integration of perspectives from multiple members of a group; and
- enabling comparison of alternative solutions to one determined music prototyping problem.

The awareness of group members plays a crucial role in supporting cooperative and multidisciplinary activities in CODES. The main aspects of these awareness mechanisms are discussed below.

The Music Prototyping Rationale Mechanism is an effective way to represent and record explanations for each action or decision made during the music prototyping process. Each user may associate comments in favour or against any action on a prototype element. The ability to associate such comments to steps in the design process is an aspect originally proposed in the human-computer interaction field, and called ‘Design Rationale’, abbreviated DR (Lee and Lai 1991). Design Rationale is a communication mechanism for the design team to communicate past critical decisions, what alternatives were investigated, and the reasons behind the chosen alternative (Moran and Carroll 1994). As an immediate consequence, it encourages deliberation and explicit consideration of alternatives. There are many models and notations for Design Rationale, like the Issue-Based Information System (IBIS) (Conklin 1998) and the Questions, Options and Criteria (QOC) notation (MacLean, Young, Bellotti and Moran 1991). Nowadays, DR is adopted also by other disciplines (like, for example, Requirements Engineering, and Systems Engineering) and recognised as a possible way to allow a group member to obtain a better understanding of other group members’ actions and decisions. Musical actions and decisions are usually subjective. It is important therefore to have a specific communication mechanism for commenting on actions, in order to inform the reasons of such actions to the other members of the group, such as the selection of a particular sound pattern, instrument, rest, etc., or the decision of making combinations or deleting some prototype element.

We consider the process of creating music or making musical experiments to be also a process that can be composed by objective decisions and choices. When the users are prototyping in CODES, they combine the musical or sound pieces in their line with other pieces from others’ lines. Thus, choices, selections, enabling, disabling and execution tasks are performed constantly in a cyclic process until an agreement about the prototyping result is achieved. All of these actions can be argued in CODES by users in order to inform the other users about the reasoning behind these actions. This is a sound way to provide awareness in asynchronous collaborative environments.

In CODES, Music Prototyping Rationale basic elements are issues and comments. Issues correspond to decisions or actions that have been made or states that have been achieved during a collaborative music prototype creation and refinement. For example, an issue can be ‘Change instrument line Y’, ‘Insert pause at time 4’ or ‘Mixing different rhythms’ (see Subject in figure 7(a)). Issues are goal-motivated consensual choices, concerning alternatives of the action course. Comments are asserted in order to support the selection of a specific course of action (comments in favour) or avert the users’ interest from it by expressing some objection (comments against). Additionally, comments may express some suggestion, question or generic observation about the issue. Comments are consensual explanation, not an individual message interchanged between actors. Every decision or action may be linked to comments (in favour or against) (see Content in figure 7(a)).

Figure 7. Screenshots of CODES comment windows.
A practical example of the Music Prototyping Rationale, after a rhythm experiment session done in the CODES environment, is described as follows. Three users participate in the same music prototype, namely Alex, Neil and Geddy. Each one of them creates one line in the prototype according to their preferred skills. However, Alex has the idea of mixing different styles of rhythms and decided to create another drum line of jazz inside a pop style. In order to obtain feedback from the other partners cooperating in this prototype, Alex writes an explanation for his action.

In the comment window, the edition of a comment includes the comment subject and the free-style body of the comment. The user can assign to a comment an optional feature, the property Private, in the case of just registering the comment for himself, because it should be invisible to others. The default value is Public, where the comments are visible to every member (figure 7(a)). Then, CODES saves the comment and associates it with the corresponding prototype issue, informing other users by a Post-it icon (see comments in figure 2) that some comment was made by some user related to some action carried out in this line. By clicking the icon, a user can retrieve the comment.

As shown in figure 7(b), our approach of Music Prototyping Rationale uses a hierarchical structure to represent the reasoning of users. Each entry in the description corresponds to an argumentation element. In that applet window each element is accompanied by three icons, one (+/-) for folding/unfolding purposes, the second (the magnifying glass) indicating that the comment was read, and the last (the sound monitor) indicating that a sound event was listened to by the corresponding user. Finally, each entry in the structure may contain its author’s username and the content of the comment. As users can manipulate the prototype at different times, access to Music Prototyping Rationale in CODES is easy and possible from any location.

Action Logging – CODES has an Action Logging mechanism to record some information (like date, author, action, prototype element affected, etc.) from all actions, making them available for all partners to observe what was done and in which sequence. The Action Logging is responsible for textually presenting a history of all actions by users via the system. When users perform actions on the prototype (login, logout, select sound pattern, add explanations, etc.), they are recorded by the system. The result is a Group Memory, a common database for all users sharing the same musical prototype and allowing users to understand not only their own activities (querying the history recorded in the group memory) but also the activities of other group members.

Figure 8 shows a sample screen of CODES, where it is possible to view an excerpt from an Action Logging organised in a chronological sequence. The user can also browse it in order to search previous actions. As shown in figure 8, the user can filter the level of information registered by events triggered by the system. By browsing, all users can track progress in the recorded trail looking for the steps that led to the current prototype state.
Modification Marks – While CODES supports long prototyping sessions, cooperative activities and prototype modifications can take place over an extended but limited period, from a few days to possibly several weeks. Under these conditions, people cooperating in music prototyping construction should have easy access to the modifications that have emerged from their activities. Thus these modifications should of course be maintained across sessions, and notification of their existence should be explicitly shown for other users. This is the main motivation for the CODES Modification Marks mechanism. As mentioned before, every time there is a change in a prototype, CODES triggers an event that is recorded automatically in the Group Memory (Action Logging). In order to indicate to a user that a prototype has been modified by others, the icon ‘N’ indicating ‘New’ appears in the line rendering.

Finally, cooperation in CODES can provide interesting alternatives for beginners in music. By means of interactions with and advice from more experienced users, CODES provides a support for beginners’ learning, a positive interdependency, encouraging collaborative actions, argumentation, discussion and cooperative learning during the development of a music prototype. This becomes more clear if analysed from the perspective of Vygotsky’s (1978) work, which 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.

7. ALTERNATIVE USES FOR CODES

This section discusses other possibilities and situations where CODES can be used. CODES has been applied in a number of small-scale studies in order to evaluate its use, identify and correct problems, and determine new requirements. The results to date express the relative success of our work, but surprisingly one of the most interesting results is the set of alternative applications the users found for CODES. In addition to the ‘conventional’ edit-listen-publish-cooperate-refine procedure for which we have initially developed CODES’ functions, users were able to find other creative ways of using it:

- as an effective support for music learning;
- as an entertainment tool; and
- as an accompaniment system for human performance.

In music learning situations, CODES can provide interesting alternatives for beginners (Miletto, Pimenta and Costalonga 2004; Miletto, Pimenta and Vicari 2005). A realistic scenario would be as follows: a collective prototype is created by a group formed by students and supervised by a music teacher. The group can carry out musical experiments leading to a music prototyping situation where each student plays a defined role and is responsible for an activity to be developed in this prototype. The group, through interactions and the advice of the teacher, decides which musical genre will be studied, as well as the number and the kind of instruments and music structures that will be put together in the prototype. Then, it is possible to work on music creation collectively, using the metaphor of a musical orchestra: each student plays a defined role in the final result. In addition, the teacher can enable several patterns related to the same instrument for different students, and all students can compare their different contributions, choosing or mixing alternatives. The teacher can also apply concepts of musical dynamics and expressiveness, indicating different sonic structures in different moments of the prototyped musical discourse. Thus, the combination of both cooperative music prototyping and music education is a promising one, and merits additional research. In particularly, our intention is to provide computer-supported features for music education following Keith Swanwick’s C(L)A(S)P model (Swanwick 1979), as we have already done in previous works (Krüger et al. 1999; Flores et al. 2001).

As an entertainment tool, CODES provides interesting experiences to users involved in musical activities like those carried out by DJs/producers (cutting/pasting metaphor). Choosing sound patterns from the system, users have the possibility of mixing different rhythms, music parts and styles just by having parallel lines of each of them playing together. Also, with a collection of added lines with sound patterns selected, the user can produce many styles of a music prototype just by enabling and disabling the groups of lines he wants, experiencing different combinations of sonority.

Another possibility is to use CODES as an accompaniment system for human performance. An individual skilled user created some lines in CODES in order to imitate the performance of a music group (a band). As she plays guitar, the lines of bass, rhythm and chords (base) were played back while she played her musical instrument. Thus, in this situation CODES is considered as an auxiliary tool for instrument practice and self-training.

As a communication tool, CODES also provides the possibility of use in music discussion forums, where users can – by means of a message board mechanism – participate in discussions and reasoning about registered ideas. An important feature of CODES is the possibility of users linking their explanations with their respective music prototypes. We are currently documenting these alternative usages of the system for a more systematic evaluation.
8. FINAL DISCUSSION

The CODES approach for cooperation among users in order to create collective music prototypes is promising, because it enables knowledge sharing by means of rich interaction and argumentation mechanisms associated with each prototype modification. Consequently, each participant may 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 down some barriers for people (especially non-musicians) who wish to engage in musical prototyping and experimentation. Of course, the musical quality of the finished work is not under discussion here, only the mere possibility of ‘creating it’.

The initial barrier to non-specialists in music is related to the possession of musical instruments. Historically, it was necessary to own or have access to an instrument to be able to investigate sound possibilities or try composing/playing sounds and music. Now, the use of multimedia computers associated with sound cards and virtual instrument software has ‘freed’ people from the need for a physical instrument to perform and compose music. We do not advocate that this is a better scenario for music making, but it is a fact, which is present reality. The objective of our work is to devise tools that take advantage of this fact in order to empower the non-specialist to create music.

Another barrier is the necessity of knowing common music notation (the score). This notation (staff and symbols) is fundamental for a deep and complete learning of musical theory, despite its complexity. However, it may pose some resistance for the layman. A non-specialist in music may consider it as an immediate practical obstacle because he needs to learn a great number of symbols and how to use them to produce musical material. Using CODES, one does not need to understand conventional music notation to create prototypes: he or she may select, play and combine sound patterns in an interactive way, without taking into account their representation format. Yet, the possibility of conversion from musical prototype to score version is one of our next goals, in order to provide more pedagogical support and understanding of musical theory.

CODES was made available for use by actual users in our academic context. Taking into account some well-known subjective evaluation methods from the HCI and Ergonomics literature (Dix, Finlay, Abowd and Beale 1998), we have employed user-satisfaction-oriented interviews as a simple evaluation procedure. In order to capture all comments and criticism (and also to avoid non-recorded opinions), users were invited to answer all questions in loco, immediately after using CODES. The interview contains few objective questions, takes only a few minutes to answer, its data analysis is quick and easy, and the whole process costs very little.

The questions were conceived to identify adequacy of information ordering and structuring for user activities during prototype creation/editing and collaboration. This includes specific items about usability, accessibility, navigation complexity, and finally if users were satisfied with the implemented functionality. In addition to the objective answers, the interview gave users the opportunity to include comments, and so, for us, an opportunity to examine these comments formally.

Preliminary results show the need for improvements with respect to some aspects of the CODES interface, such as:

- providing a better time control, enabling the user to stop at any time and resume playback from that time;
- allowing listening to the sound pattern before entering it into the timeline;
- allowing users to upload their own sound patterns into the lines of CODES; and
- providing the standardisation to represent musical information in the GUI.

An aspect remaining to be explored is the implementation of real-time interaction in the CODES architecture. Interaction among users in a synchronous mode may bring up new features and properties not considered until now.

The ultimate goal of our work is to make CODES available for public use, to expand our audience and achieve our intent of providing universal access to the experience of the creative act. As people interested in music experimentation become more sophisticated, tools supporting this process need to employ more efficient technologies and approaches in order to address their complexity and diversity. The present work suggests that cooperative music prototyping may become an increasingly important topic in this regard.

ACKNOWLEDGEMENTS

The authors would like to thank Eduardo Reck Miranda (Future Music Lab, University of Plymouth, UK) for his advice on this research project and comments on this paper. This work was partially supported by the Brazilian research funding councils CNPq and CAPES.

REFERENCES


