A Hybrid Approach to Recommend Partners in Collaborative Musical Environments

Guilherme G. Testa[1], Marcelo S. Pimenta[1], Damián Keller[2], Victor Lazzarini[3]

[1]Instituto de Informática – Federal University of Rio Grande do Sul (UFRGS) – Porto Alegre – RS – Brazil

[2]NAP – Núcleo Amazônico de Pesquisa Musical - Federal University of Acre (UFAC) – Rio Branco – AC – Brazil

[3]National University of Ireland, Maynooth, Ireland

{ggtesta,mpimenta}@inf.ufrgs.br, dkeller@ccrma.stanford.edu, victor.lazzarini@nuim.ie

Abstract. *Collaboration is an important element in the human creative construction process, and takes special significance in music composition. Systems that support collaborative activities with creative focus have been growing largely lately. However, there is a lack of tools that assists users of these systems to find partners to collaborate with in their activities. In this context, this paper presents a hybrid approach – based on both content-based and collaborative filtering – to recommend partners in creative collaborative environments focused on their produced work. Its main characteristics and functionalities are demonstrated by its design on CODES, a platform for music prototyping for lay users.*

1. Introduction

Collaboration is a key concept in the human creative process. Michael Farrel, in his extensive study about collaborative circles, mentions a passage of Henry James where he declares that "every man produces more when it is in company of people working in the same field of work" [Farrel 2003]. Farrel complements saying that collaborative circles affect significantly the creative work. Artists, writers, composers and politicians reported that cooperation is an indispensable part for the developing of their works. Several important studies of sociology, anthropology and psychology have demonstrated the importance of collective creativity to the intellectual and technological advancement of human society [Palispis 1996][Winthrop 1991][Tuomela 2000].

In general, collaborative processes to build products that are not generated by methodological steps require special support for collaboration. Creative activities tend to feed from mutual inspiration between its collaborators: ideas flow around the discussion of elements that composes the creative work in the pursuit of a satisfactory outcome for all (which is not always known *a priori*). This means that all parts must be tuned to the final results and having similar expectations. But at the same time, it is desirable that every person could offer new and innovative ideas to contribute in a unique manner to the final result.

In this sense, conventional approaches to stimulate cooperation in the design of technical products, where the objectives and responsibilities are usually well defined, do not allow a systematic and opportunistic trading of ideas [Pimenta et al. 2011]. Therefore, they are not suitable for the dynamic and creative nature typically associated with collaboration among musicians. In this sense, a new concept of awareness is needed to define this new form of collaboration [Miletto et al. 2006]. Similarly, we believe that the recommendation of partners in non-technical byproducts should incorporate some main differences to optimize the suggestion of compatible partners.

In this context, we present an approach for recommending people (partners) for collaborative musical environments. This proposal will be demonstrated from its implementation and usage in CODES, a collaborative music prototyping environment focused on people with no prior musical experience [Miletto 2010]. Our recommendation approach is hybrid, combining both content-based and collaborative filtering. Thus, characteristics of users are recovered to create musical profiles, so they can be compared and analyzed for partner recommendation.

This article is organized as follows: section 2 presents an overview of CODES. In section 3, we discuss some related works and studies. Section 4 explores the details of the recommendation system being proposed. Section 5 describes the user experience of the partner recommendation in CODES. Finally, section 6 presents some conclusions and future work.

2. Exploring CODES

CODES is a Web-based environment designed to support Cooperative Music Prototyping (CMP), built with special focus on music novices. The main focus of CODES is to approximate the lay user to music creation, without a need of theoretical music concepts. CODES offers a high level music representation and user interface features to foster easy direct manipulation of icons representing sound patterns (predefined MP3 samples with 4 seconds of duration).

Built with appropriate support features, CODES users can create, edit, share and publish simple musical pieces - or Musical Prototypes (MPs) collaboratively to a restricted group or to the Web. These shared MPs can be repeatedly listened to, tested, and modified by the partners who cooperate on prototype refinement. Users can start a new MP just by choosing the name and dragging the musical instruments to the editing area. Edition in CODES includes intuitive actions to build a MP. It is possible to "drag-and-drop" sound patterns from the sound library, "move", "organize", "delete", "expand" the duration, and "collapse" to listen to the final result. When sharing a musical prototype, the "owner" user can invite other users to listen and collaborate with his prototype, or may send explicit invitations via e-mail to non-members asking them for cooperation. When someone accepts such an invitation, the user becomes a prototype partner and can edit the MP like the owner does.

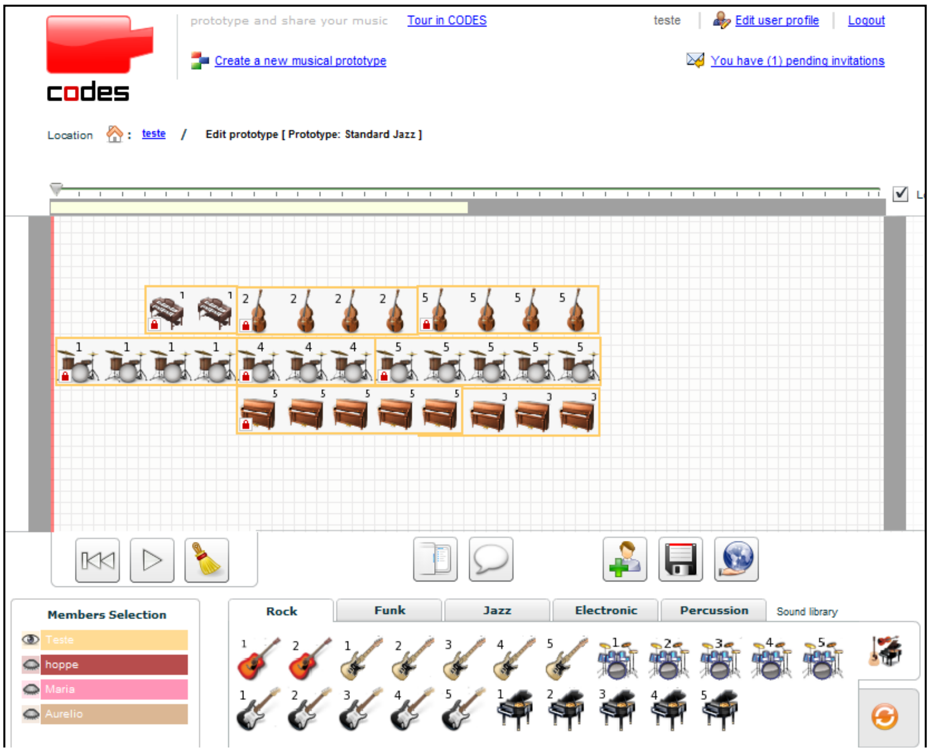

Figure 1. Editing a MP in CODES

The prototypical nature of CODES is designed and built to provide a novice-oriented perspective. All the interactions with the editing area are made as intuitive as possible, without any music theory terms. All prototype partners can discuss and change ideas about each step of the prototype refinement, in order to understand each other's decisions. It is possible to link arguments to every decision made. 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 can export their musical prototype, and share it at will.

2.1. Awareness Mechanisms in CODES

Through CODES, ordinary users may have the opportunity to be the actors of their own musical experiences. Also, the system offers tools to allow a full cooperation among partners. This implies a focus not only on community management but also on experimenting and participating in specific design practices using a suitable interaction vocabulary. This process suggests the existence of noteworthy distinct kinds of cooperation activities. Awareness and conflict resolution are already considered critical issues in general Computer Supported Cooperative Work (CSCW). However, mechanisms existing in other systems need some adaptation to take into account the idiosyncrasies of the CMP context. The ultimate goal is to provide actual cooperation, social knowledge construction, argumentation and negotiation among the actors of the MP design activities. This type of cooperation is supported by a set of mechanisms borrowed from the Software Engineering and Human-Computer Interaction (HCI) areas and specially adapted for CODES [Miletto 2010].

2.2. Design and Development of CODES

The design and development of CODES adopted a user-centered and incremental approach, taking into consideration social aspects such as the characteristics of the users, contexts, purpose, minimal technology requirements, and the nature of its possible influence on the novice user.

CODES is based on the classical client-server architecture for Web applications. In the current version of CODES, special attention was given to aspects related to interaction flexibility and usability since one of the main goals is to implement an adequate support for manipulation of complex musical information, cooperative activities and group awareness, to provide an effective interaction of the users with each other and with the environment itself. Thus, in the client-side, CODES uses scripts embedded within standard HTML (Figure 1).

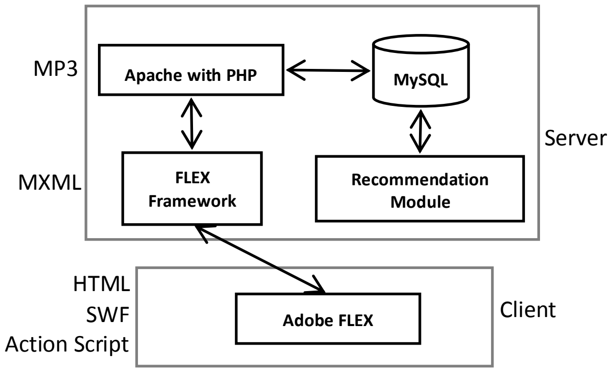

Figure 2. CODES current architecture

On the server side, CODES implements the Model-View-Controller (MVC) architecture. 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). The Adobe Flex as script language 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, while the HTML 5 standard is still being on development [W3C 2013]. The sound files used in CODES are small MP3 files which can be quickly downloaded by the client-side ensuring a standard audio quality.

The recommendation module connects to the MySQL database to extract information about the users and to write the data structures of the recommendations. This communication does not interfere in the response of the server to the client.

3. Related Work

There are several studies that approximate partially to the work being proposed in this paper. CODES and it recommendation system grabs several researches areas as

Recommendation Systems, Sound and Music Computing (SMC), Music Recommendation, Computer Supported Cooperative Work (CSCW) and so on.

To introduce the analysis of these areas of study, we will present, in the next subsections, important works classifying them into three main areas: people recommendation, music recommendation and collaborative environments of music composition.

3.1. People Recommendation

Although recommendation of general items is a problem already well researched both in academia as in business and commercial environments, it is not possible to assert the same about people recommendation. In CODES, our focus is to recommend other users (partners) to collaborate with in the process of music prototyping. Recommending individuals is a difficult task due to the subtle inherent and instinctive aspects of the human relations, especially in creative activities. Studies in this area are growing, mainly focusing on social networks.

In music, a person has greater empathy with people with a similar musical taste. This assertive actually can be propagate to several other areas of interest. When more things in common we have with another individual, more we tend to sympathize and relate better with him [Guy et al. 2010][Guy et al 2009]. The key point here is similarity. Briefly, the problem of recommending people is to construct a profile that reflects faithfully the main characteristics of a user, and then, compare those profiles to look for similarities.

Therefore, to make a good recommendation it is necessary an extra effort than just the application of the usual recommendation algorithms. These differences of recommending people instead of products became evident with the growth of virtual communities, and the direct application of the collaborative filtering technique in such systems demonstrated to be ineffective [Cai et al. 2010]. Individuals differ in ways that the standard algorithms are not adequate to present a good recommendation [Lopes et al. 2010]. It is necessary to adopt further measures to encompass all these details.

In this sense, CODES differs from the recommendation engines usually designed to recommend people because our focus is partnership. People recommendation is mainly driven by virtual communities centered on personal relationships (new friendships, for example), or for technical activities. In a creative environment like CODES, characteristics as empathy are not essentials for a good partnership. Otherwise, is very important that the collaborators have the same goals and are also able to work together efficiently.

3.2. Music Recommendation

Several interesting academic studies about music recommendation can be found in the literature, using different focuses. In generic terms, music recommendation deal with the problem of recommending a list of songs that pleases a certain user. To achieve that, it is necessary to understand the elements that construct the musical taste of a user, including as many details as possible, to feed a personal musical profile of the user. In this sense, several information can be used to enrich this profile, like the explicit searches of the user, his reviews or feedback about songs listened, favorites songs and artists, his personal profile and others [Chen and Chen 2001][Shao et al. 2009][Bogdanov et al. 2011].

With all this information, another question reveals to be crucial: the automatic analysis of the music themselves. To categorize songs, artists and genres efficiently, systems extract semantic information from audio files and use it to group similar artists and songs. Algorithms that provide sound analysis require a considerable amount of processing, but they are becoming viable with the advancements of hardware. Websites that implements these techniques are growing and becoming popular on the Internet and people are changing for listening downloaded music to streaming media as Grooveshark and Pandora

However, almost all of those websites are focused on simple playback of tracks and are not related to music creation or collaboration itself. Also, the focus of recommendation in CODES is not audio analysis, once the information about the genre and style are already explicit in the system.

3.3. Collaborative environments for music composition

Collaborative environments for music composition are also growing with the increase of interactive community websites. Some interesting services are already available on the Internet today, and they have a significant number of users, which demonstrates the growing potential of this area.

Some of them are Indaba, Kompoz and SoundCloud. All these provide a platform for publication, storage and support collaboration in different ways. In Indaba, the user can also make use of an interesting tool to edit and save directly voice or instrument sounds using a microphone connected to the computer. Kompoz is focused mainly in collaboration itself, and it is possible to upload and publish recordings of musical material so that other users can use freely. Sound Cloud, differently, serves as a repository of sounds and songs.

CODES, unlike, focuses in users who have no previous knowledge of music theory and, in this sense, it differs from the examples cited. Additionally, none of these websites provides recommendations to bring users with similar musical tastes together for musical composition.

4. Recommendation in CODES

There are two approaches widely used in recommender systems in general: content-based and collaborative filtering. The first uses representations of data that were accessed in the past to create user profiles. Based on these profiles, the system then recommends new items for a specific user that are possibly relevant, according to his activity history [Balabanović and Shoham 1997]. In collaborative filtering, similarities between the actions of different users are analyzed to predict a behavior. Users with similar profiles are grouped together to help the recommendation. Collaborative filtering is based on the premise that people who historically have similar activities, are more likely to behave similarly also in the future [Sarwar et al. 2001].

Looking to take advantage of the main benefits of both techniques of recommendation, some hybrid algorithms were created mixing characteristics of these two methods. In a hybrid approach, we use information about both the history of user activities as its similarity with other users to search for compatible profiles. The recommendation can also count with other relevant information to refine the similarity calculation [Kim 2008][Cai et al. 2009].

In CODES, as the user interact with the system, the database is being populated with relevant information that is used by the recommendation engine to produce a musical profile. CODES has a sound library with several predefined samples of instruments of different genres. A combination of these sounds patterns produces a MP. The number of occurrences of different instruments and different genres are collected of the range of MPs created by the user to approximate the preferred genre for composition of a user. Then, with the musical profile created, similarity calculation takes place to achieve recommendation.

The musical profile of the user is represented by a matrix that is filled while the user composes MPs. These matrixes are them combined and compared with other parameters to mold the final profile. The initial genre matrix is populated as follows.

$$ncomp(p)_u = \sum_{i=0}^{m} nmp_{i,u} \times 0,9^i \Big/ \sum_{i=0}^{m} 0,9^i \tag{1}$$

Where in (1), $ncomp(p)_u$ is the combined grade of the prototypes of the user p calculated separately for each genre u. nmp represents the value of the genre u for a specific music prototype, while i corresponds to the order of the MPs of the user, being 0 the most recent to the oldest m. Therefore, each MP will have a 10% decrease in the relative score of its immediate precedent. The sum of all the music prototypes of the user as their respective genres will result in a matrix of genres that takes into account a large number of compositions to quantify the musical preference of a user.

In order to avoid costly calculations that can compromise the scalability of the system, it is preferable to assign a fixed value to calculate the genre matrix for each user [Agrawal and Srikant 1994]. In this case, considering that the compositions themselves are the predominant factor to draw the music profile, this maximum value should not be too low. For this reason, it is initially assumed $m = 20$. The genre matrix is updated by the profile manager to each new MP edited or created by the user through the collaborative prototyping module.

4.1. Evaluation of public items

The recommender system in CODES implements a well-known rating system to evaluate the opinion of the user to any item. This explicit information is an important way to get a user feedback about others' MPs. It helps to reflect interests and rejections that may go unnoticed by analyzing only the contributions of a single user.

CODES has a rating system of public MPs that follows a standard widely used in evaluations of items, with grades assigned from 1 to 5. The impact of an evaluation alters the influence of a single MP positively or negatively in the total of the public MPs calculated. Table 1 lists the relative influence of each grade.

Table 1. Relative influence according different grades.

Grade assigned	Opinion associated	Relative influence
1	Terrible	-0.25
2	Bad	-0.1
3	Neutral	0
4	Good	$+0.1$
5	Great	$+0.25$

For each user, it is constructed an evaluation matrix that stores information about the ratings given by the user to public MPs. Initially, the matrix is completed with equally values distributed to each genre. As the user evaluates MPs, the relative influence of each MP (according to Table 1) of the range of all MPs that the user gave feedback is calculated and the table updated. This matrix is constructed as follows:

$$\boldsymbol{natr(p)_{i,u} = natr(p)_{i-1,u}} \times (\mathbf{1} + \boldsymbol{nmp(p)_u} \times \boldsymbol{feedback}) \quad (2)$$

In (2), $natr(p)_{i,u}$ corresponds to the grade given by the user p relative to the genre u to all public MPs that the user gave feedback until i. Initially, $natr(p)_{i,u}$ has a value equally distributed among all genres; and $feedback$ represents the value of the relative influence column on Table 1 according to the grade assigned.

4.2. Clustering and similarity calculation

Clustering is an interesting way to reduce the universe of comparisons needed to build a recommendation. First, the information about the user activity history and the public evaluations of data items collected in the previous steps are combined. Then, the users are grouped according to a defined criterion and, finally, the similarity calculation is conducted, where users are actually chosen for the recommendation.

The genre and evaluation matrixes are unified using a weighted arithmetic mean (3), where each grade assigned to a public MP will have an effect of 5% of the final grade, up to a limit of 30%. I.e. the user's own MPs correspond to at least 70% of the final grade. This difference in weights seeks to give greater emphasis to the genres of MPs made by the user himself.

$$nfinal(p)_u = \begin{cases} \dfrac{(20-n)\times ncomp(p)_u + n \times natr(p)_u}{20} & if\ n < 6 \\ 0.7 \times ncomp(p)_u + 0.3 \times natr(p)_u & if\ n \geq 6 \end{cases} \quad (3)$$

In (3), $nfinal(p)_u$ represents the weighted final score for each of the genres u for the user p. $ncomp(p)_u$ refers to the relative grade of the genre u in the genre matrix of the user p. And finally, $natr(p)_u$ corresponds to the given grade of the genre u in the range of all MPs that the user gave feedback (as explained in section 4.1).

After set the unified genre matrix, the clustering part takes place. Each user is assigned to a specific group according to the majority of its genres of interest. This group separation is intended to reduce the universe of calculations needed for comparison, making the system faster and more scalable [Cataltepe and Altinel 2007]. In CODES, clustering is made according to each user's favorite genres, and these genres are those who contribute, jointly, with at least 80% of the total number of MPs in the unified genre matrix of the user.

The next step is the similarity calculation. One of the most used methods to calculate the proximity between users in recommend systems is the Cosine Similarity (or Vector Similarity), which presents an interesting measure for comparison and prediction [Vozalis and Margaritis 2003]. Note that the summations are calculated over a number of items for which both users u and p have expressed their opinions. The correlation then is calculated as follows:

$$dist_{u,p} = \sum_{i=1}^{n} \frac{r_{i,u} * r_{i,p}}{\sqrt{\sum_{i=1}^{n} (r_{i,u})^2} \sqrt{\sum_{i=1}^{n} (r_{i,p})^2}} \quad (4)$$

In (4), $r_{i,u}$ is the rating of the user u for the corresponding item i i, in the n-dimensional item space. If the user didn't rate the current item, the value is set with 0.

A similarity or neighborhood matrix, S, can now be generated, including the similarity values between all users of a given group. The entry at the x-th row and the y-th column corresponds to the similarity between two given users, u_x and u_y.

In possession of S, is possible to recommend the n closest users of the current user consulting the neighborhood matrix. The distance of the acceptable users for recommendation is dynamic, being evaluated and refined through the user feedback about the previous recommendations. As said in the introduction, in creative activities, sometimes people are not looking for a similar user to interact but, unlike, for someone that can contribute with different ideas. This creates a recommendation area, which is shaped with an increase or decrease of 10% of the similarity value for each evaluation of a collaborator recommended by the system.

The list of users of the recommendation area for a specific user u is stored in the user profile itself, and is updated regularly. This communication intends to keep the processing and time consumption of the similarity calculations transparent to the user, considering that the total amount of calculations for the whole recommendation procedure can be costly.

5. User Experience

As explained in the previous section, the process of recommendation is transparent to the user. A list of suggestions about potential partners is showed in three different situations: in the main page, in the summary page and in the music prototyping edition page.

In the first situation, the user invites a recommended partner to start a new MP. The user writes an invitation detailing his intentions, so the potential collaborator can analyze and decide to accept it or not. If the user denies a suggestion, it will vanish from the list of potential collaborators. Otherwise, if the user accepts and collaborates to prototype a MP, this user is allocated to a list of accepted recommendations. Both users then can leave a feedback about the mutual experience, and this feedback will be used to refine the next recommendation procedure.

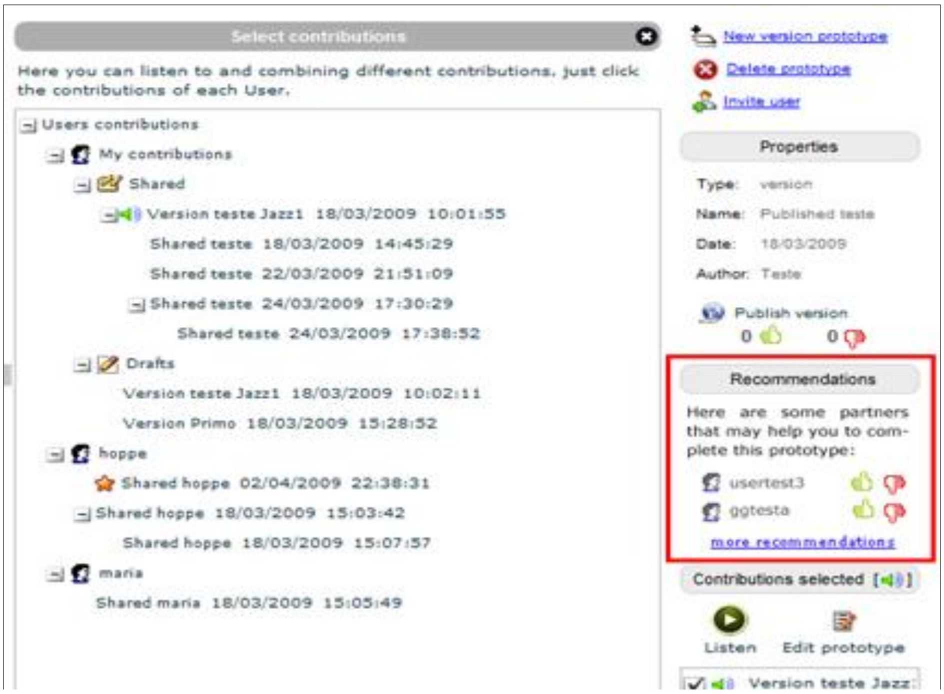

Figure 3. Editing a MP in CODES

In the other two situations, the users will be invited to collaborate in a MP that is already being prototyped (by one or more users). In these cases, the MP is attached to the invitation, so the invited user can listen to the current MP that he is being invited to collaborate with, and can have a better decision.

6. Evaluation

A multi-criteria evaluation of the recommender system proposed in this article is in phase of development, which means that aren't preliminary results of it. The main focus of this evaluation is to get user feedback about the recommendations of partners in CODES, and to adjust eventual needs. The evaluation will be conducted in 5 steps, as follows:

1. Prototyping: a set of 20 different people without prior knowledge about music composition will be invited to create at least 5 music prototypes in CODES, respecting their own musical taste;
2. Evaluation: the set will be indicated to give feedback (rate) of at least 5 others' MPs;
3. Recommendation: with the data collected in stages 1 and 2, the recommendations will be followed by the participants;
4. Cooperation: the users will be guided to create at least 2 MPs with each of the top 2 user recommended;
5. Evaluation: a final questionnaire will be presented to the participants to evaluate the collaborative composition and the recommendation of the users.

This evaluation will be very important to measure the effectiveness of the recommender system presented, as its capacity to give good recommendations. Also it will be an important source to guide future revisions or changes in the recommendation approach.

7. Conclusions

In this paper we presented a hybrid approach to recommend partners in a musical collaborative environment. First, we introduce inherent factors that influence partnership work in collaborative creative activities, such as mu-sic. Then we proceeded with a brief presentation about CODES – a platform for music prototyping for lay users –, and an explanation about the recommender engine projected and being implemented in CODES.

Although the preliminary results of the system evaluation with real users have not being collected at this stage, we believe that a hybrid approach concerned and intended to include the subtle aspects of partnership in creative activities is the more suitable approach to make recommendations in collaborative musical environments.

8. References

Agrawal, R. Srikant, R. (1994) "Fast Algorithms for Mining Association Rules in Large Databases". VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc. San Francisco, CA.

Balabanović, M. Shoham, Y. (1997) "Fab: content-based, collaborative recommendation". Magazine of Communications of the ACM. CACM Homepage archive. Volume 40 Issue 3. ACM New York, NY. 1997.

Bogdanov, D. Haro, M. Fuhrmann, F. Xambo, A. Gomez, E. Herrera, P. (2011) "A content-based system for music recommendation and visualization of user preferences working on semantic notions". 9th International Workshop on Content-Based Multimedia Indexing (CBMI). 249-252.

Cai, J. Francis, J. Gheysens, S. (2009) "Creating a Hybrid Music Recommendation System from Content and Social-Based Algorithms". GSET '09, Governor's School of Engineering and Technology.

Cai, X. Bain, M. Krzywicki, A. Wobcke, W. Sok, Y. K. Compton, P. Mahidadia, A. (2010) "Collaborative Filtering for People to People Recommendation in Social Networks". AI 2010. Lecture Notes in Computer Science. Volume 6464/2011, 476-485.

Cataltepe, Z. Altinel, B. (2007) "Music recommendation based on adaptive feature and user grouping". 22nd International Symposium on Computer and Information Sciences, ISCIS„07. 1-6.

Chen, H. Chen, A. L. P. (2001) "A Music Recommendation System Based on Music Data Grouping and User Interests". CIKM '01 Proceedings of the tenth interna-tional conference on Information and knowledge management. ACM New York, NY, USA.

Farrel, M. P. (2003) "Collaborative Circles: Friendship Dy-namics and Creative Work". University Of Chicago Press (Nov. 2003). ISBN 978-0226238678. 328p.

Guy, I. Jacovi, M. Perer, A. Ronen, I. Uziel, E. (2010) "Same places, same things, same people? Mining user similarity on social media". Proceedings of the 2010 ACM conference on Computer Supported Cooperative Work. 41-50. ACM New York, NY.

Guy, I. Ronen, I. Wilcox, E. (2009) "Do you know?: recommending people to invite into your social network". IUI '09: Proceedings of the 14th international conference on Intelligent User Interfaces. ACM New York, NY.

Kim, K. Lee, D. Yoon, T. Lee, J (2008). "A music recommendation system based on personal preference analysis". First International Conference on the Applications of Digital Information and Web Technologies, ICADIWT„08. 102-106.

Lopes, G. R. Moro, M. M. Wives, L. K. Palazzo, J. M. O. (2010) "Collaboration recommendation on academic social networks". ER'10 Proceedings of the 2010 international conference on Advances in conceptual modeling: applications and challenges. 190-199.

Miletto, E. M. (2010) "CODES: an Interactive Novice-oriented Web-based Environment for Cooperative Musical Prototyping". Ph.D. Thesis. PPGC, Instituto de Informática, UFRGS.

Miletto, E. M. Flores, L. V. Rutily, J. Pimenta, M. S. (2006) "CODES: supporting awareness in a web-based environment for collective music prototyping". In Proceedings of VII Brazilian symposium on Human factors in computing systems (IHC '06). ACM, New York, NY, USA, 50-55.

Palispis, E. S. (1996) "Introduction to Sociology and Anthropology". Manila: Rex Printing Company, Inc. ISBN 971234875X. 313p.

Pimenta, M. S. Miletto, E. M. Flores, L. V. Hoppe, A. (2011) "Cooperative mechanisms for networked music". Future Generation Computer Systems archive. Volume 27, Issue 1, January, 2011. 100-108. Amsterdam, The Netherlands.

Sarwar, B. Karypis, G. Konstan, J. Riedl, J. (2001). "Item-based collaborative filtering recommendation algorithms". WWW'01 Proceedings of the 10th international conference on World Wide Web. ACM New York, NY, USA.

Shao, B. Wang, D. Li, T. Ogihara, M. (2009). "Music Recommendation Based on Acoustic Features and User Access Patterns". IEEE Transactions on Audio, Speech, and Language Processing, Vol. 17, No. 8.

Tuomela, R. (2000) "Cooperation: A Philosophical Study". Kluwer Academic Publishers, Netherlands. ISBN 0-7923-6201-2. 431p.

Vozalis, E. Margaritis, K. G. (2003) "Analysis of Recommender Systems Algorithms". HERCMA‟ 03 Hellenic European Research on Computer Mathematics & its Applications Conference. 732-745, Athens, Greece.

W3C. (2013) "HTML 5". Available on http://www.w3.org/html/. 2013.

Winthrop, R. H. (1991) "Dictionary of Concepts in Cultural Anthropology". Greenwood Press, New York. ISBN 0-313-24280-1.