Recommending audio mixing workflows

Sauer, Christian, Roth-Berghofer, Thomas, Auricchio, Nino and Proctor, Sam (2013) Recommending audio mixing workflows. In: 21st International Conference on Case-Based Reasoning, 8-11 Jul 2013, Saratoga, USA.

Recommending Audio Mixing Workflows

Christian Sauer, Thomas Roth-Berghofer, Nino Auricchio, and Sam Proctor

School of Computing and Technology, University of West London,
St Mary's Road, London W5 5RF, United Kingdom
{first.lastname}@uwl.ac.uk

Abstract. This paper describes our work on *Audio Advisor*, a workflow recommender for audio mixing. We examine the process of eliciting, formalising and modelling the domain knowledge and expert's experience. We are also describing the effects and problems associated with the knowledge formalisation processes. We decided to employ structured case-based reasoning using the *myCBR 3* to capture the vagueness encountered in the audio domain. We detail on how we used extensive similarity measure modelling to counter the vagueness associated with the attempt to formalise knowledge about and descriptors of emotions. To improve usability we added GATE to process natural language queries within *Audio Advisor*. We demonstrate the use of the *Audio Advisor* software prototype and provide a first evaluation of the performance and quality of recommendations of *Audio Advisor*.

Keywords: CBR, myCBR, audio, mixing, audio engineering, similarity measures, workflow recommendation, knowledge formalisation

1 Introduction

With automatic composition and improvisation of music expressing the individual style of a human composer as well as the automatic expressive performance of music, the two main steps of music creation are quite well researched [15,16]. There are a variety of approaches to automated composition of expressive music and the expressive performance of music, see e.g., [15,4]. They all need to deal with the problem of formalising emotions in order to relate to the intended emotional effect of a composition and/or performance. The formalisation of affective, emotional statements or descriptive adjectives of an emotion is still a problem [10,6], often encountered by applications dealing with art and deeply linked to emotions and perception of such.

Next to composition and performance, a third important task in professional music production is the mixing of a sound recording. Mixing is the process of applying a set of spectral modifications to sounds in order to achieve a change in timbre or more specifically the emotional effect of the sound on a listener [13]. This process is goal-oriented, with the goal being a desired change in the emotional effect of a sound. The vocabulary describing this effect-change consists of terms that describe the emotion desired to be triggered or altered, i.e., increasing or decreasing an emotional effect. We find queries like 'make it sound more warm' or 'make it sound less harsh' and onomatopoeia in the language of audio engineers.

The experience of audio engineers is in the linkage between queries containing timbre descriptors such as 'warm' or 'bright' as well as amount descriptors and constrains, and in the choice and application of spectral modifications used to achieve the desired timbre change of the sound. Additionally the effect of such a query is also linked to the context in which it occurs. The modelling and (re-)use of such context embedded queries to recommend the adequate workflows was the main goal of our *Audio Advisor* prototype workflow recommender.

This paper introduces our work on *Audio Advisor*, a workflow recommender system that allows its users to formulate natural language queries for the automatic case-based retrieval of workflows that, when applied to the audio product changes its timbre and/or applies an effect to it. The workflow itself is provided as a sequence of so called presets, where a preset can be described as a selection of frequency descriptors with definite decibel change values for said frequencies. A preset can further contain information on defined effects such as *reverb* or *delay* and the decibel values to be applied to these effects. An example is to define a preset to reduce high frequencies and emphasise lower frequencies while adding a slight echo effect to the sound.

The rest of the paper is structured as follows: We interlink our approach with the current state-of-the-art in the field of artificial music composition and performance in Section 2. Based upon the goals and aims of *Audio Advisor* (Section 3) we examine the domain of audio mixing and its specific knowledge as well as our approaches to elicit and formalise the knowledge in Section 4. In Section 5 we show how we use *GATE*[1] to develop the natural language processing component that enables *Audio Advisor* to 'understand' natural language queries posted to the system. We then demonstrate how we use *myCBR 3*[2] for *Audio Advisor* and examine the overall structure and workflow of the *Audio Advisor* application. Section 6 details on the performed experiments regarding the quality of the workflow recommendations and evaluate the performance of *Audio Advisor*. A summary and outlook on future work then concludes the paper.

2 Related work

A variety of approaches to formalise emotional annotations and/or descriptive terms that either describe the mood of the music or the way it is to be played [16,14] already exists. Such approaches deal with either playing music in a certain defined way to convey an emotion [7] or to select songs or sounds that are associated with a mood or emotional state [21]. For automated composing, the question of integrating a formal description of the mood the composed music should match is already well researched [16,4].

Emotions or, in our context, the timbre of a sound and its perception are not easy to be a) defined and b) quantised/formalised [12,8,10]. Another problem

[1] http://gate.ac.uk/
[2] http://www.mycbr-project.net

we were facing during the domain knowledge formalisation was that we tried to quantify and cluster descriptive adjectives based on very vague data given by the individual descriptions of the emotional effect a sound has on a person describing this effect. The difficulties of capturing a sounds timbre are [9]: "*It is timbre's 'strangeness' and, even more, its 'multiplicity' that make it impossible to measure timbre along a single continuum, in contrast to pitch (low to high), duration (short to long), or loudness (soft to loud). The vocabulary used to describe the timbres of musical instrument sounds indicates the multidimensional aspect of timbre. For example, 'attack quality, 'brightness,' and 'clarity' are terms frequently used to describe musical sounds.*" The vagueness of the data is based on said variation in the individuals perceptions when they either should describe an emotional effect or perceive something that is annotated with a particular emotion but have a complete different idea of the actual emotion this percept triggers [17,9,11].

The problems caused by the described vagueness of timbre descriptors, which we initially examined in [20], were one of the most prominent ones during the knowledge formalisation process employed for the *Audio Advisor* application's knowledge model. We were able to counter said vagueness by employing complex similarity measures, following the knowledge modelling procedures described for example in [1]. Additionally we also investigated how to extract the meaning, thus the semantics of a natural language query posted to our *Audio Advisor*. We did so mainly by following an approach we developed for a previous Information Extraction (IE) application, *KEWo*, that extracts taxonomies of terms to be used as similarity measures in CBR systems from natural language texts [19].

3 Aims and Opportunities of our work

A way to circumvent the lack of quantifiable measures and vagueness is to allow for vagueness and a certain amount of ambiguity within the techniques used for formalising and retrieving problem descriptions based on descriptive adjectives. The vagueness accompanying the formalisation of these descriptive adjectives, mainly the timbre descriptors, can be handled by the use of similarity knowledge in Case-Based Reasoning (CBR) systems [2,22]. The ability of CBR to handle said vagueness has already been used to guide the emotional component of automatic composition as well as performance of music, see e.g., [7,3,18,16].

The aim of *Audio Advisor* is to make audio mixing experience available to its users and to allow them to use and learn the special vocabulary employed by experienced audio engineers. By making the audio engineers experience available through our *Audio Advisor* we thus are able to fulfil the following goals:

- Using *Audio Advisor* in a teaching approach for audio engineering students
- Allowing lay persons to practise / improve mixing skills
- Re-use the knowledge of experienced audio engineers
- Speed up the mixing process and, thus, reduce expensive studio time
- Improve usability to audio mixing software by integrated workflow recommendations

4 The Audio Mixing Domain

The most common mixing task is to change an input sound and consequently an *input* timbre to a desired *target* timbre by a specified *amount*. This basic problem description can be extended by a number of sub timbres to be changed simultaneously and constraints on the desired changes such as 'Make the flute sound more airy but not so breathy.'. Following this basic assumptions about the audio mixing domain, we present in this section our approach to elicit the domain knowledge from experienced audio engineers as well as the knowledge artefacts we were able to elicit. We then consider the problems we faced during the knowledge formalisation process and their influence on our choice of the formalisation techniques that we employed. We then review our resulting initial knowledge model that we modelled using *myCBR 3* and which is currently used as the reasoning component of the *Audio Advisor*.

4.1 Domain Knowledge

The knowledge representing the experience of audio engineers has a high grade of abstraction and is highly encoded. For example the knowledge how to apply a set of frequency changes in a specific order to change a sound in a specific context with a desired effect is simply encoded in a sentence like: 'Make the trumpet a lot fatter and a bit more toppy, like in Jazz music'. This sentence is implicitly associated by the experienced audio engineer with a workflow like: *Increase the 6 kHz frequency in the high shelf segment by 3 dB, then increase the 150 Hz segment by 9 dB with a wide bandwidth and finally reduce the 2.7 kHz segment by 2 dB with a narrow bandwidth.*

Due to this high level of abstraction and encoding we faced the problem of choosing the best suited techniques for the necessary knowledge elicitation. We opted for employing a variety of techniques to minimise the danger of knowledge loss and to maintain a high level of accuracy. To get insight into the audio mixing domain we arranged for several studio sessions where the audio engineers provided actual hands on experience on how to mix an audio product in a studio. Second to these sessions we arranged for a couple of interview sessions with two audio engineers. During these interviews we questioned the experts so they could provide their experience in increasing grades of formalisation.

The knowledge elicitation process also yielded some unexpected artefacts. For example, the audio engineers came up with Venn diagrams classifying the timbre descriptors. Such artefacts were very helpful while building the taxonomic similarity measures for the timbre descriptors.

4.2 Initial Knowledge Modelling

After the elicitation of the described knowledge artefacts that describe a mixing task the next step was to design an initial knowledge model of the audio mixing domain. As we already stated in section 1 one of the most complicated challenges while trying to formalise descriptors for timbres, is the vagueness of said descriptors. We decided to counter this challenge by employing structured CBR as we expected to counter the vagueness of the timbre descriptors and amount descriptors by modelling complex similarity knowledge that describes their relationships. We quickly identified the main domain relationship, presets being applied to timbres, as a perfect candidate to divide the domain into a problem and solution part

The most foreseeable challenge we encountered was the challenge of finding an optimal grade of abstraction. This was of importance as we were, like in any knowledge formalisation task, facing the trade-off between an over engineered too specific knowledge model and the danger of knowledge loss by employing too much abstraction e.g. choosing the abstraction levels too high. Together with the domain experts we chose two additional abstraction levels of frequency segments for the timbre descriptors. We further chose to use a taxonomic order for the timbre descriptors and the amount descriptors, as well as the instruments to be used as structures to model the respective abstraction layers of these knowledge artefacts. Thus we designed taxonomies describing timbres, amounts and instruments from a most abstract root node down to the most specific leafs, see our initial work on this approach [20] for details.

The next modelling step consisted of determining the best value ranges for the numerical attributes we wanted to integrate into our initial knowledge model. Again after discussing this with the domain experts we agreed to use two way to represents *amounts* in our domain. We provide a percentage approach, ranging from 0 to 100% as well as a symbolic approach. The symbolic approach was chosen because the domain experts mentioned that from their experience the use of descriptors for amounts, such as *'a slight bit'* or *'a touch'* were by far more common in audio mixing sessions then a request like *'make it 17% more airy'*. So we integrated, next to the simple and precise numerical approach, a taxonomy of amount descriptors into our initial knowledge model. The taxonomy was ordered based on the amount the symbol described, starting from the root, describing the highest amount down to the leaf symbols describing synonyms of smallest amounts. Additionally to modelling the amounts we also needed to represent the workflow steps, so the application of presets. For this we elicited that the application of spectral modification's is always specified in decibels (dB) and that these settings always follow a certain rasterization, due to the knobs and dials on a mixing board clicking into place with certain amounts of dB being tuned in on this dials. We thus provided the amounts in the workflow descriptions in decibel.

Regarding *myCBR 3* we had to choose between a taxonomic and a comparative table approach. Considering the versatile use of taxonomies in structural CBR [5] we initially opted for the use of taxonomies. Yet regarding the complex similarity relationships between the elicited timbre descriptors we also wanted to investigate whether a comparative table approach for modelling the similarities of the timbre descriptors might yield a more accurate knowledge model, ultimately resulting in better workflow recommendations. So we formalised the similarities of the timbre descriptors also using the comparative table approach.

5 Prototype implementation

In this section we will detail on how we implemented the *Audio Advisor* application prototype using the GATE framework and *myCBR 3*.

5.1 Using GATE for Natural Language Query Processing

Audio Advisor allows a user to enter a natural language queries such as 'Make the trumpet a bit brighter but not too airy.' Such a query requires the *Audio Advisor* to be able to parse the natural language into settings for the attribute values of the mixing task's problem description. Figure 1 shows the automatically set attribute values based on a real sample query.

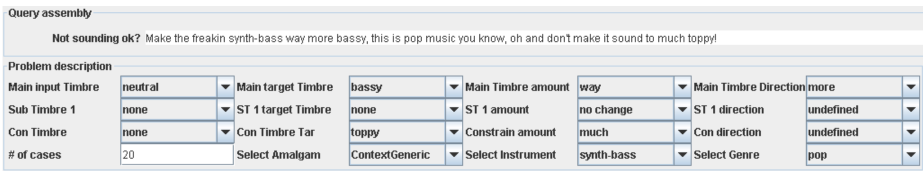


Fig. 1. Problem description section of the *Audio Advisor* application GUI

To extract the correct attribute values and their context from the natural language query we employ the GATE Architecture, i.e., a modified version of the ANNIE application[3]. To, for example, distinguish between a query with and without a constraint, we analysed the structure of a number of example queries with the use of the GATE Developer 7.0 GUI application, see Figure 2 for details. After designing the necessary specially built language processing resources, i.e., Gazetteers and Jape grammar rules, we modified the ANNIE Application to allow for the Annotation of the following term categories: amount, constraint, direction, effect, instrument, timbre and timbre-shift. By using these annotations we were able to analyse the query structure as the following figure demonstrates:

The structural analysis of the queries enabled us to build a classification tree that represents typical semantics formulated in a certain type of query. In this way we can map the queries to reoccurring kinds of problem descriptions and set the values specified within the query to the correct attributes describing specific mixing tasks. Figure 3 shows a section of the classification tree.

The customised ANNIE is embedded in the *Audio Advisor*. The annotations generated by the customised ANNIE application are stored in an XML file that is then parsed to make the annotations available for the query assembly.

[3] http://gate.ac.uk/sale/tao/splitch6.html

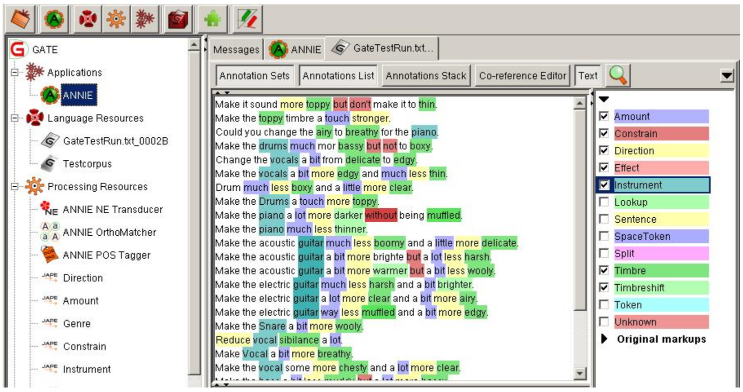


Fig. 2. Query Annotation in GATE

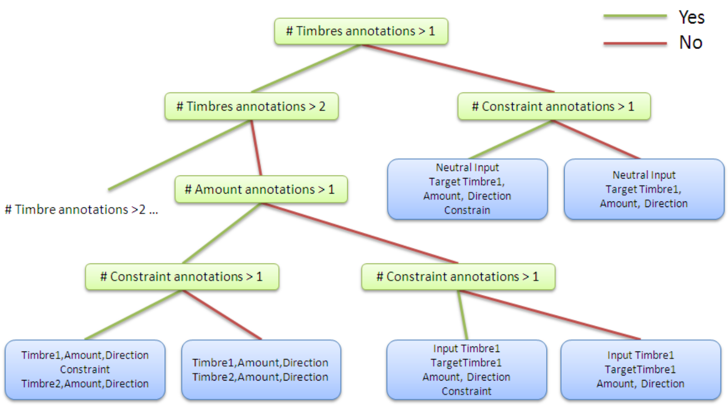


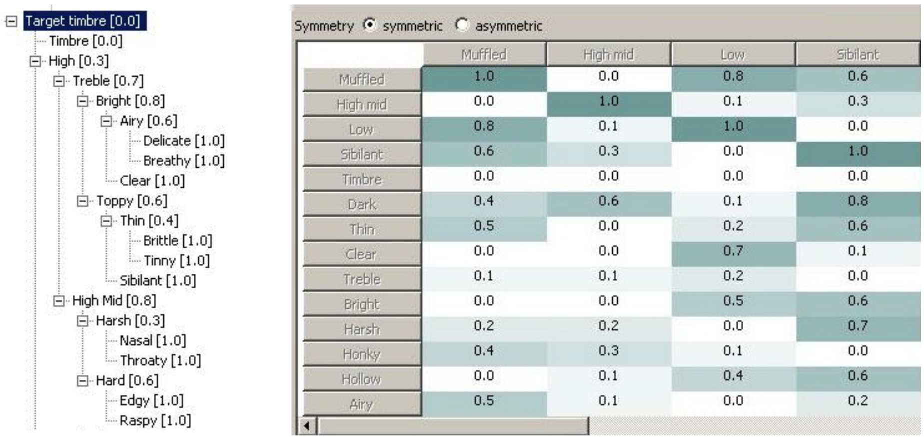
Fig. 3. Classification tree derived from query analysis (excerpt)

5.2 CBR Engine modelling

myCBR 3[4] provides the knowledge engineer with a variety of graphical user interfaces that allow for rapid prototyping of CBR knowledge models. We used *myCBR 3* Workbench to swiftly transfer our initial knowledge model into a structured CBR knowledge model. Figure 4 provide an insight in the modelling of the local similarity measure for timbre descriptors. The first figure shows the taxonomic modelling on the left and a section from the same similarity measure being modelled in a comparative symbolic table on the right.

The problem description consists of two attributes, *MainInputtimbre* and *SubInputtimbre1* describing the current sound. Additionally the problem description contains the two Attributes *MainTargettimbre* and *SubTargettimbre1* that are used to specify the timbres into witch the sound should be changed. The

[4] http://www.mycbr-project.net



	Muffled	High mid	Low	Sibilant
Muffled	1.0	0.0	0.8	0.6
High mid	0.0	1.0	0.1	0.3
Low	0.8	0.1	1.0	0.0
Sibilant	0.6	0.3	0.0	1.0
Timbre	0.0	0.0	0.0	0.0
Dark	0.4	0.6	0.1	0.8
Thin	0.5	0.0	0.2	0.6
Clear	0.0	0.0	0.7	0.1
Treble	0.1	0.1	0.2	0.0
Bright	0.0	0.0	0.5	0.6
Harsh	0.2	0.2	0.0	0.7
Honky	0.4	0.3	0.1	0.0
Hollow	0.0	0.1	0.4	0.6
Airy	0.5	0.1	0.0	0.2

Fig. 4. Timbre descriptor taxonomy

attributes *TimbreAmountEffectDescriptor*, *AmountDirectionDescriptor* and optionally *TimbreAmountPercentage* are available for the Main as well as the Sub timbre. They are used to describe the amount of change that is intended to take place and its direction e.g. if the timbre should be increased or decreased. The last two attributes of our case structure are a String, *WorkflowDescription*, which holds a String that is describing the workflow necessary to achieve the desired timbre change described in the Problem description part. Additionally the Case holds the original natural language query on which it was modelled as a reference. Regarding our initial case base we were able to elicit 30 commonly encountered audio mixing tasks from our domain experts. Based on these 30 initial cases we conducted a series of retrieval tests together with our domain experts.

5.3 *Audio Advisor* at work

In the following we will give a brief overview over the process of recommending a mixing workflow by *Audio Advisor* (cf. Figure 5). Upon starting the application it initialises a new instance of the Gate framework using the GATE embedded functionalities. Into this instance the customised ANNIE Application is loaded and initialised *Audio Advisor* then uses the *myCBR 3* API to initialise a new mycbr3 project. Loading the case base into the CBR project finishes the initialisation sequence of the *Audio Advisor* and the program is ready to be used.

Input to the *Audio Advisor* application can be provided by either manually selecting values from the drop down menus available in the problem description section of the *Audio Advisor* GUI, or by entering a natural language query into the 'Query Assemble' text field (see Figure 1). If a user decides to enter a natural language query the text is copied into the corpus of the ANNIE application and processed. The ANNIE application returns annotations of timbre

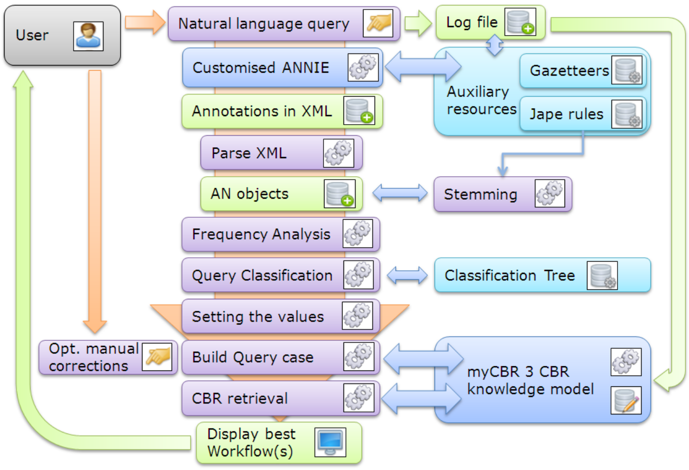


Fig. 5. Recommendation process diagram

descriptors, amount descriptors, timbreshifts, amount directions and constraints in an XML file, which is then read and unmarshalled by an instance of the QueryExtractor class. Within this class the annotations are analysed regarding the frequency of certain types of annotations, e.g. how many timbres are annotated or if there are annotations present annotating a constraint. Based on this analysis the classifier tree is searched for a query structure best matching the query characteristics (annotation frequencies: Number of timbres, number of amount descriptors, presence of constraint annotations) to identify the most likely structure of the natural language input query. Based on the best identified query structure the drop down menus of identified attributes are populated with the extracted values. The user can always adjust the values manually and/or add additional values. By clicking the 'Recommend workflow' button the user triggers the recommendation process.

The GUI of the *Audio Advisor* is quite straight forward (Figure 6). The upper part of the GUI provides all the elements necessary for a user to specify the audio mixing problem at hand. Additionally the user can select which amalgamation function should be used for retrieval. This allows retrieval of mixing tasks in the context of different genres. The query is then analysed and a workflow is recommended by a similarity based retrieval within the CBR Engine. The resulting order of best matching audio mixing workflows is then presented to the user in the lower section of the GUI.

6 Experiments and Evaluation

In this section we explain the aims and setup of the experiments we performed with *Audio Advisor*.

Our knowledge elicitation effort led us to the following knowledge artefacts:

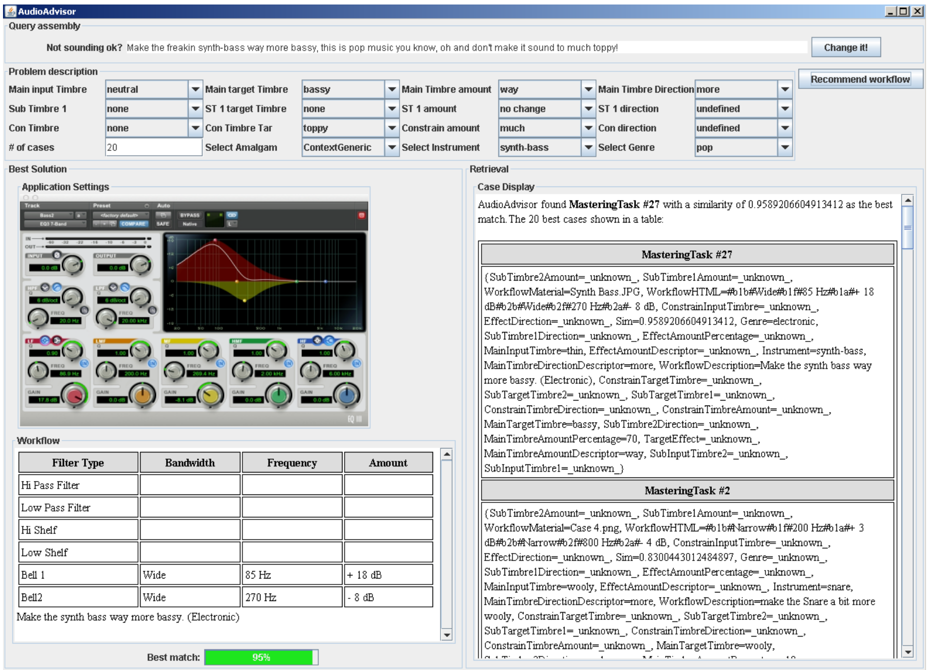


Fig. 6. *Audio Advisor* reading a natural language query containing a constraint

- o A set of 39 timbre descriptors with varying grades of abstraction
- o A set of 21 amount descriptors
- o A set of 15 Direction descriptors
- o A set of 20 Effect descriptors with varying grades of abstraction
- o Similarity of timbre descriptors in taxonomic and comparative table form
- o Similarity of amount descriptors in taxonomic and comparative table form
- o Similarity of amount descriptors as Integer function (distance function)
- o Similarity of Context of Genre in taxonomic form
- o Similarity of Context of Instrument in taxonomic form
- o Similarity of the Effect descriptors in taxonomic form
- o Global Similarities of the problem description of the mixing task depending on the selected Genre and Instrument context
- o 30 Screenshots of application settings of the used mixing software
- o 30 initial cases describing 30 common mixing tasks

On these artefacts and the knowledge model consequently modelled from them, we performed experiments to establish the quality of the knowledge model. The main goal of the experiments was to gain an insight into how good our approach to formalise experience, from anecdotal into fully formal, worked with regard to avoiding knowledge formalisation problems. We further aimed to evaluate the performance/quality of our Software prototype working with our initial knowledge model. Our third goal was to establish the usability and applicability of our overall approach of workflow recommendation in day to day audio mixing work and teaching scenarios. Our fourth goal is it to establish if the use of taxonomies or the use of comparative symbolic tables yields more accurate similarity measures and thus better recommendations.

6.1 Setup of the Experiments

We performed two series of experiments. The first series aims at establishing the usability, quality of recommendation and performance of the *Audio Advisor* application in the day to day use of the software by experienced audio engineers. The second series of experiments aims at establishing the usability of the *Audio Advisor* application for teaching audio engineering students and gather feedback on the quality of the application's recommendations.

At the current time we have conducted experiments from the first series and are currently preparing experiments for the second series. The setup for the first experiment was the following: Two experienced audio engineers were asked to use the *Audio Advisor* software to enter natural language queries into it describing common audio mixing tasks. The engineers were to provide feedback on the usability of the recommended workflows. They were also asked to provide feedback on the correctness of the similarity ordering or sequence of the 5 best matching cases that were retrieved. The data gathering for this experiment was accomplished by logging the natural language queries the audio engineers entered into *Audio Advisor* as well as by providing the audio engineers with questionnaires to provide us with their feedback on *Audio Advisor*'s workflow recommendations. The questionnaire asked for the description of problems encountered with the retrieved workflow, for example, not being applicable for a certain instrument. Further the questionnaire asked for a rating of the quality, applicability of the recommended workflow ranging from 1 (worst) to 5 (best). The third information we gathered was the comparison of the case sequence retrieved to the case sequence deemed optimal by the audio engineers.

The second series of experiments will also use questionnaires to gather feedback from audio engineering students. Students will bring in their own work, consisting of sound samples and songs which they still need to optimise and use the *Audio Advisor* application to get recommendation on how to do so. They will then employ these recommendations on their work (sounds) and rate the actual outcome with regard of the extent the sound has changed as it was intended by the student. Additionally the students are asked for feedback on how fast, in terms of iterations of: Entering query, retrieve workflow, apply workflow in studio, they deem a learning effect to set in. This estimated learning effect will be verified by audio engineering lecturers in the form of a small practical test.

Both series are planned to be repeated with an improved knowledge model that will use symmetric symbol tables as similarity measures rather than the taxonomies used in the first place. This repeated series of experiments aims at providing us with data to compare the performance and accuracy of the two knowledge formalisation approaches we employed.

6.2 Evaluation

Here are first results from our first series of experiments. Each audio engineer was asked to enter 10 queries and provide us with feedback on the applicability of the recommended workflow and the sequence of the first 5 most similar cases retrieved by the *Audio Advisor*. To provide an idea of what kind of natural languages queries were entered by the engineers here is a short excerpt from the actual *Audio Advisor* log file: 'Can you make the drums more toppy?, 'Make the drums more toppy.', 'Change the bass to be more bassy but not toppy.', 'Make the flute more airy but not breathy.', 'The drums need to be way more heavy but not to boomy for a pop song.'

As stated before, the questionnaire we used, asked for the description of problems encountered with the retrieved workflow, for example, not being applicable for a certain instrument. Further the questionnaire asked for a rating of the quality, applicability of the recommended workflow ranging from 1 (worst) to 5 (best). As an informal kind of feedback both engineers reported that if the recommendation was above their rating of 2 it usually was quite useful and perfectly applicable. The third information we gathered was the comparison of the case sequence retrieved to the case sequence deemed optimal by the audio engineers. Table 1 lists the aggregated feedback from both audio engineers with regard to the similarity of the best retrieved case to their query in per cent and the applicability of the recommended workflow:

Table 1. Ratings of results

Rating by Audio Engineer 1	Best	Worst	Average
Match of query case to best retrieved case	95%	69%	79%
Applicability of workflow [1:worst to 5:best]	4	1	2.7

Rating by Audio Engineer 2	Best	Worst	Average
Match of query case to best retrieved case	98%	57%	77%
Applicability of workflow [1:worst to 5:best]	4.5	1	2.13

The second kind of data we gathered from our first set of experiments was the sequence of the five best cases, sorted in a descending order based on their similarity to the query posted. Additionally to his retrieved sequence of cases we asked the audio engineers to provide us with their ordering of the cases and respective mixing workflows with regard of their applicability to the query the engineer entered into the system. We did so to get an insight into the quality of the similarity measures with regard to their effect of "prioritising' the sequence of workflows to recommend in an accurate order. Accurate order meaning the first recommendation (case) being the most applicable and then have the "next best solution" and the 'next best' and so on in a sequence of decreasing applicability. Out of the 20 queries tested 5 were retrieving optimal sequences, the remaining

15 sequences are shown in the following table 2 displays the case sequences with the retrieved sequence in the top row and the engineers suggested optimal sequence in the lower row and starting with the best cases being on the left side of the table:

Table 2. Case retrieval sequence comparisons

	Engineer 1						**Engineer 2**				
Retrieved sequence	11	3	1	4	7		1	4	2	7	24
Optimal sequence	1	4	11	3	7		24	7	2	4	1
Retrieved sequence	2	7	27	3	1		0	17	12	15	21
Optimal sequence	7	2	27	3	1		17	0	12	21	15
Retrieved sequence	2	1	7	27	4		3	2	1	7	4
Optimal sequence	2	7	27	1	4		4	7	3	2	1
Retrieved sequence	11	26	6	25	13		3	2	1	7	4
Optimal sequence	26	13	11	25	6		2	3	7	1	4
Retrieved sequence	11	3	1	7	2		13	20	26	6	23
Optimal sequence	3	1	11	7	2		20	26	6	13	23
Retrieved sequence	2	1	7	4	11		12	15	17	0	21
Optimal sequence	2	7	1	4	11		17	15	12	0	21
Retrieved sequence	3	11	2	1	7		3	11	4	2	1
Optimal sequence	2	7	3	11	1		4	2	11	3	1
Retrieved sequence	/	/	/	/	/		9	23	13	26	14
Optimal sequence	/	/	/	/	/		9	13	23	26	14

Overall, next to the 25 % of optimal retrieved sequences, the remaining 75% retrieved sequence's orderings were labelled as of sufficient quality by the audio engineers and as a good basis for suggesting alternative workflows. The engineers reported still some flaws in detecting certain amount descriptors. Additionally sometimes the separation of Maintimbre and Subtimbre was not correctly extracted from the natural language query. Both engineers reported that the case structure and interface might still be reduced more to only Input timbre Target timbre an amount descriptor and a constraint on timbre. Overall extraction of the queries from the natural language input was rated as usable, except for the explicitly reported shortcomings which are due to our not yet refined Gazetteers and Jape rules we employ in our ANNIE IE application. The overall feedback from the engineers was quite enthusiastic as they reported to us that once we refine the knowledge model slightly further and made minor changes to the query extraction the *Audio Advisor* actually would be quite powerful in supporting audio mixing and the teaching of audio mixing.

7 Summary and Outlook

In this paper we presented our development of a case-based workflow recommendation system for audio engineering support. We detailed on the entire process of developing the *Audio Advisor* software. We described our approach to formalise the special vocabulary, consisting of vague descriptors for timbres, amounts and directions. We introduced CBR as a methodology to amend the problem of formalising emotions and/or adjectives describing timbres, i.e., the problem of the vagueness of terms and the variance of emotions invoked by the same sound in different humans. We further detailed on our approach to design a Case-based reasoning knowledge model based on the elicited knowledge artefacts. We then described how we designed and implemented the *Audio Advisor* application. While doing so we inspected the *GATE*based natural language processing ability that we integrated into *Audio Advisor* to enable it to process queries posted to it in natural language. We further detailed on the use of *myCBR 3* to rapidly prototype and refine the CBR knowledge model that poses the reasoning component of the *Audio Advisor*. We finished this paper with an overview of our experiments with the *Audio Advisor*and an introduction to a first evaluation of the performance of the *Audio Advisor* and the quality of its recommendations which overall are very promising in both possible roles of the *Audio Advisor* as a support tool for professionals as well as a teaching aid to students.

For the imminent future we plan to refine our knowledge model further based on the evaluated data from the first series of experiments. The next step is the conduction of the teaching related experiments and the further refinement of the knowledge model as well as the GUI employed by *Audio Advisor*.

As a medium future aim we want to investigate the possibility and usefulness of using 'negative similarity values' of the timbre descriptors provided by the domain experts during the knowledge elicitation phase. We therefore want to integrate a negative similarity measure into our knowledge model and perform experiments to establish as this might be useful to be employed as adaptation knowledge as we already suggested in our initial study of the formalisation of knowledge from the audio mixing domain.

References

1. Aamodt, A.: Modeling the knowledge contents of CBR systems. In: Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning. Citeseer (2001)
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 1(7) (Mar 1994), `ftp://ftp.ifi.ntnu.no/pub/Publikasjoner/vitenskaplige-artikler/aicom-94.pdf`; letzte Verifikation 11. Juni 2007
3. Arcos, J., Grachten, M., de Mántaras, R.: Extracting performers behaviors to annotate cases in a cbr system for musical tempo transformations. Case-Based Reasoning Research and Development pp. 1066–1066 (2003)

4. Arcos, J., De Mantaras, R., Serra, X.: Saxex: A case-based reasoning system for generating expressive musical performances*. Journal of New Music Research 27(3), 194–210 (1998)
5. Bergmann, R., et al.: On the use of taxonomies for representing case features and local similarity measures. In: Proceedings of the 6th German Workshop on Case-Based Reasoning. pp. 23–32 (1998)
6. Broekens, J., DeGroot, D.: Emotional agents need formal models of emotion. In: Proc. of the 16th Belgian-Dutch Conference on Artificial Intelligence. pp. 195–202 (2004)
7. Canamero, D., Arcos, J., de Mántaras, R.: Imitating human performances to automatically generate expressive jazz ballads. In: Proceedings of the AISB99 Symposium on Imitation in Animals and Artifacts. pp. 115–20. Citeseer (1999)
8. Darke, G.: Assessment of timbre using verbal attributes. In: Conference on Interdisciplinary Musicology. Montreal, Quebec (2005)
9. Donnadieu, S.: Mental representation of the timbre of complex sounds. Analysis, Synthesis, and Perception of Musical Sounds pp. 272–319 (2007)
10. Fellous, J.: From human emotions to robot emotions. Architectures for Modeling Emotion: Cross-Disciplinary Foundations, American Association for Artificial Intelligence pp. 39–46 (2004)
11. Halpern, A., Zatorre, R., Bouffard, M., Johnson, J.: Behavioral and neural correlates of perceived and imagined musical timbre. Neuropsychologia 42(9), 1281–1292 (2004)
12. Hudlicka, E.: What are we modeling when we model emotion. In: Proceedings of the AAAI spring symposium–Emotion, personality, and social behavior (2008)
13. Katz, B., Katz, R.: Mastering audio: the art and the science. Focal Press (2007)
14. de Mántaras, R.L.: Towards artificial creativity: Examples of some applications of AI to music performance. 50 Anos de la Inteligencia Artificial p. 43 (2007)
15. de Mantaras, R.: Making music with AI: Some examples. In: Proceeding of the 2006 conference on Rob Milne: A Tribute to a Pioneering AI Scientist, Entrepreneur and Mountaineer. pp. 90–100 (2006)
16. de Mantaras, R., Arcos, J.: AI and music: From composition to expressive performance. AI magazine 23(3), 43 (2002)
17. Pitt, M.: Evidence for a central representation of instrument timbre. Attention, Perception, & Psychophysics 57(1), 43–55 (1995)
18. Plaza, E., Arcos, J.: Constructive adaptation. Advances in Case-Based Reasoning pp. 306–320 (2002)
19. Sauer, C., Roth-Berghofer, T.: Web community knowledge extraction for myCBR 3. In: Research and Development in Intelligent Systems XXVIII: Incorporating Applications and Innovations in Intelligent Systems XIX Proceedings of AI-2011, the Thirty-first SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence. p. 239. Springer (2011)
20. Sauer, C., Roth-Berghofer, T., Auricchio, N., Proctor, S.: Similarity knowledge formalisation for audio engineering. In: Petridis, M. (ed.) Proceedings of the 17th UK Workshop pn Case-Based Reasoning. pp. 03–14. University of Brighton (2012)
21. Typke, R., Wiering, F., Veltkamp, R.: A survey of music information retrieval systems (2005)
22. Watson, I.: Case-based reasoning is a methodology not a technology. Knowledge-Based Systems 12(5), 303–308 (1999)