European and Mediterranean Conference on Information Systems 2009 (EMCIS2009)

July 13-14 2009, Crowne Plaza Hotel, Izmir

ONE’S OWN SOUNDTRACK: AFFECTIVE MUSIC SYNTHESIS

Atta Badii, Intelligent Media Systems and Services Research Laboratory (IMSS), School of Systems Engineering, University of Reading, UK
atta.badii@reading.ac.uk

Ali Khan, Intelligent Media Systems and Services Research Laboratory (IMSS), School of Systems Engineering, University of Reading, UK
a.a.khan@reading.ac.uk

David Fuschi, Intelligent Media Systems and Services Research Laboratory (IMSS), School of Systems Engineering, University of Reading, UK
d.l.a.fuschi@reading.ac.uk

Abstract

Computer music usually sounds mechanical; hence, if musicality and music expression of virtual actors could be enhanced according to the user’s mood, the quality of experience would be amplified. We present a solution that is based on improvisation using cognitive models, case based reasoning (CBR) and fuzzy values acting on close-to-affect-target musical notes as retrieved from CBR per context. It modifies music pieces according to the interpretation of the user’s emotive state as computed by the emotive input acquisition componential of the CALLAS framework. The CALLAS framework incorporates the Pleasure-Arousal-Dominance (PAD) model that reflects emotive state of the user and represents the criteria for the music affectivisation process. Using combinations of positive and negative states for affective dynamics, the octants of temperament space as specified by this model are stored as base reference emotive states in the case repository, each case including a configurable mapping of affectivisation parameters. Suitable previous cases are selected and retrieved by the CBR subsystem to compute solutions for new cases, affect values from which control the music synthesis process allowing for a level of interactivity that makes way for an interesting environment to experiment and learn about expression in music.

Keywords: Affective, Emotive, Music synthesis, Emotion, CALLAS.

1 INTRODUCTION

The ability to interact with a subject depends to an extent on the level of empathy that can be reached. This holds in most contexts, but is particularly relevant in the case of Human Machine Interaction. A successful film, opera, theatre piece, album or any other multi-media object bears an associated aural-memory. There are songs or tunes that characterise our memories of a certain event (be it happy, sad, pleasant or unpleasant) and this is because rhythms and audio information are extremely powerful conveyors of emotions, memories and feelings. If it is true that often a picture is worth a thousand words, we could say that a melody is worth a *full* sensorial memory, so strong, and complete, it involves and activates all senses (olfactory and visual aspects of the environment). Audio has also often been used to support therapy in many contexts given its recognised effect on people’s mood. Thus it would be justified to say that if one could create one’s own soundtrack and dynamically adapt it to the current mood; this would be a valuable result per se.

2 RELEVANT WORK

Following musical rules, no matter how sophisticated and complete, may not be enough to render added expression to otherwise mechanical sounding computer music. The challenges in deploying virtual affective musicians is to engineer an effective way of using that tacit expressive knowledge that human musicians gain and use in improvising to affectivise when playing a score; this is the sort of personal “touch” that gives the music a particular targeted quality and is about knowing when and where not to play a note or add an ornamental note. There has been significant interest in emotionally organised music collections in the past. Previous work along these lines exists in the form of the following.

SaxEx (Arcos et al. 1998) uses CBR and fuzzy logic to generate expressive performances of melodies based on examples of human performances. For musical improvisation, the Jazz Improvisation Generator (JIG) (Grachten, 2001) and SICOM (Pereira et al. 1997) have both made attempts at getting a machine to solve this task. The interactive possibilities of SaxEx were developed in NOOS language, which allows the user to choose among a variety of alternative choices that can invoke a desired affect on the resulting musical expression. SaxEx system models human musician’s affective performance expertise on a saxophone and re-uses it in delivering a targeted affective quality in the thus improvised music. The interactive possibilities of SaxEx allow the user to choose among a variety of alternative choices that can invoke a desired affect on the resulting musical expression. For example the users can express their preference along three affective dimensions: "tender-aggressive", "sad-joyful", and "calm-restless".

SICOM (Pereira et al. 1997) creates music using CBR and stores it as a MIDI file with two tracks, a soprano and bass line to produce the final result. The cases in the system represent notes in the track and the system uses CBR to decide which notes would fit best with the previously played notes. The main issues seem to be its small case library (of 3 cases) and the efficiency of its retrieval mechanism (Pereira et al. 1997). The Affective Remixer (Chung & Vercoe, 2006) arranges musical collections through the use of short clips (5 – 45 seconds) to gather emotional information. Emotional data is captured via the use of GSR (Galvic Skin Response), foot tapping (to show the level of user engagement) and subjective evaluation data (forms filled in by the user regarding their likes and/or dislikes).

3 CALLAS PROJECT

CALLAS (Conveying Affectiveness in Leading-edge Living Adaptive Systems) is an integrated project (EC FP6) that aims to reduce the cost and the complexity pertaining to the development of multimodal interfaces in digital entertainment and information context by:

- producing a set of components which can be used to generate emotionally-aware affective multimodal user interfaces
- handling interoperability between the components through a framework
- providing interfaces to third party developers through which the components and emotional model can be accessed.

Hence, CALLAS is based on a three-layer structure that maps the general objectives into operational areas. *The CALLAS Shelf* is a library of multimodal components developed and made available by the project partners, starting from state-of-the-art technologies, improved and transformed into exploitable components. *The CALLAS Framework* is a plug-in architecture for the interoperability between the components, allowing multimodal applications developers to combine them at design time, providing significant cost reduction as well as quality of software improvement. *The CALLAS Showcases* are experimental applications, significant test-beds in the context of New Media, embryonic samples of future applications, through which the capabilities of the CALLAS Framework are intended to be demonstrated.

4 AFFECTIVE MUSIC SYNTHESIS (AMS)

In this project, we are developing a Shelf component for affective output called Affective Music Synthesis (AMS) that enables the dynamic interactive modification of the characteristics of music on the basis of the user's mood. The component, at the time of writing, is able to process a MIDI file altering some key characteristics (pitch, tempo and velocity) in response to the changing mood of the user in terms of its psychological correlates as expressed in the form of Pleasure-Arousal-Dominance Model (PAD). It works in batch-mode as well as real-time to affectivise a collection (or a set) of musical pieces. Ongoing work intends to add functionality to the component whereby it would be able to generate a musical piece from a repository of pre-recorded samples that could be further affectivised either individually or as an ensemble once arranged together.

The system is based on improvisation using cognitive models, case-based reasoning and fuzzy values acting on close-to-affect-target musical notes as retrieved from the case-based repository per context. It modifies the music pieces according to the interpretation of the user's emotive state represented by the Pleasure-Arousal-Dominance (PAD) model as computed by the emotive input acquisition components of the CALLAS framework. The PAD model representation of the user's emotive state characterises the criteria for the music affectivisation process. Using previous instances of affective adaptations from a case repository, a selection of music samples are adapted according to a user's mood. The PAD values are used by the case-based reasoning subsystem for selection and retrieval of this previously stored contextually suitable case in the case-based reasoning repository. Using combinations of positive and negative states for the affective dynamics along three affective axes (*pleasure, arousal* and *dominance)*; the octants of temperament space (Mehrabian. 1987, 1991, 1996) are used as reference emotive states of the user. This provides a level of interactivity that makes way for an interesting environment to experiment and learn about expression in music.

4.1 Background

4.1.1 Case Based Reasoning

Case Based Reasoning (CBR) allows problem solving by recalling the manner in which similar problems were solved previously. The system is given a set of basic cases and a set of rules by which it can alter these cases. When given a problem, the software would find the most relevant case (or maybe multiple cases) and modify it to better solve the problem. By employing a case based reasoning approach and providing AMS with base cases incorporating mapped affectivisation parameters, the system is able to adapt to a large part of the affective input domain in the following steps:

- The input arrives from one or more shelf input components via the CALLAS framework. As an example, this might contain information that the user is 70% happy and 30% calm.
- The closest matching case is chosen and modified accordingly to become the solution. The solution contains information about which rules to follow for modifying music.
- Music is selected from the repository according the solution's instructions.
- The selected music is modified using musical rules according to the solution's instructions. This forms the final affective musical output.
- The output is analysed for effectiveness and the solution is stored in the case repository.

4.1.2 Pleasure-Arousal-Dominance (PAD) Model

The Pleasure-Arousal-Dominance (PAD) Emotional Model (Mehrabian. 1987, 1991, 1996) categorises emotions in eight base states with many intervening states:

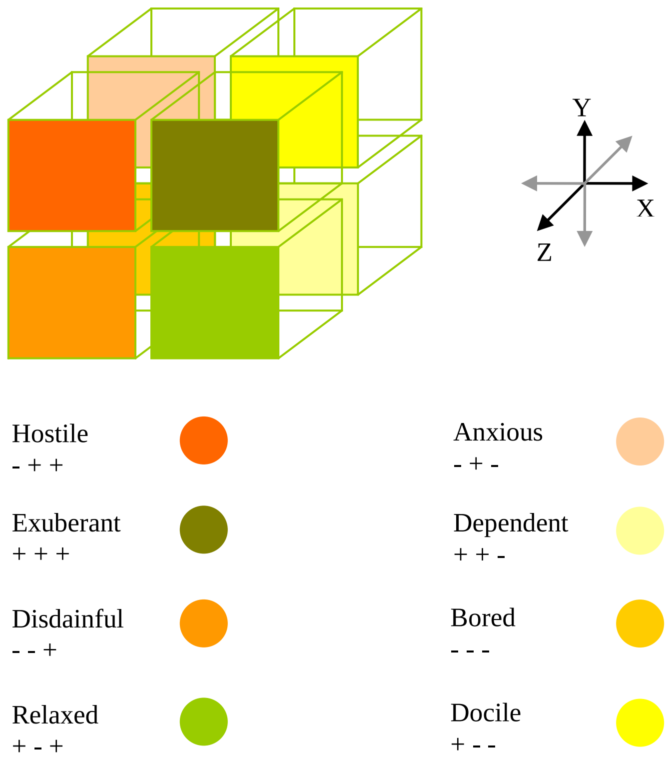

Figure 1. Octants of temperament space

Picard et al. (2001) back this idea from PAD that there are eight emotional extremes and describe the difficulty behind machine recognition of human emotion. This model was chosen by the CALLAS project partners to represent emotive state data that would be communicated among shelf components via the framework.

4.2 AMS: Design and Implementation

AMS contains three sub-systems; the Case Based Reasoning (CBR) system, the Affective Music Improvisation Engine (AMIE) and the Analysis Engine. As shown in figure 2, the first point of entry for the PAD Values (acquired by the multimodal input modules of CALLAS) into the Affective Music Synthesis module is the CBR subsystem. These PAD values reflect the emotive state of the user and represent the criteria for the music affectivisation process. The PAD values are used by the CBR subsystem for selection and retrieval of a suitable previously stored case in the CBR repository (a music sample and the parametric values of its associated solution which are to be used for modification of the sample under consideration). The solution of the retrieved case is passed to the Affective Music Improvisation Engine (or AMIE) which uses the values from the solution to affectivise the given music sample. To help improve the CBR system's selection of cases, the Analysis Engine uses further input PAD values to modify the selected case and store the solution in the repository.

4.2.1 CBR Subsystem

The CBR system contains four classes which provide the various required functionalities. A*MSRepository* stores a collection of AMSCases and offers access to these cases via a variety of methods e.g. addToRepository, removeFromRepository, selectClosestCase, generateCaseFromPAD, saveRepository and importRepository. *AMSCase* represents a case with a problem definition (the PADModel) and a solution (the MusicSample). AMSCases can also be given a name and a success rate. *PADModel* stores the three values of a PAD model and provides methods for checking the equality of instances. *MusicSample* stores the location of a music sample, and three values (Tempo, Pitch and Volume) which can be used as information about the music sample or the alterations to make.

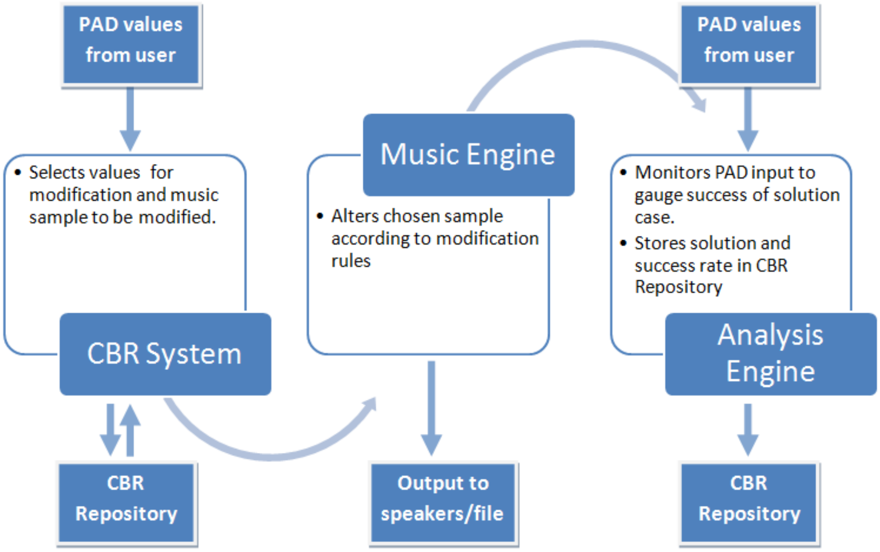

Figure 2. Affective Music Synthesis Design Overview

An example of a saved repository file can be seen below. The file contains a number of AMSCase elements within a single AMSRepository element. Each AMSCase element then has a PADModel element representing the problem and a MusicSample element representing the solution. These elements then have attributes which describe their state. Names for each AMSCase can also be stored as an attribute of the AMSCase.

PADModel elements have 3 attributes (Pleasure, Arousal and Dominance) all of which must contain a numerical value between -1.0 and 1.0. MusicSample elements contain 4 attributes. The SamplePath attribute contains a path to a music file to be played (either a relative path, or a full path). The Pitch, Tempo and Volume attributes contain the modifications which are to be made to the music.

```
<?xml version="1.0"?>
<AMSRepository>
  <AMSCase Name="Exuberant">
    <PADModel Pleasure="1" Arousal="1" Dominance="1" />
    <MusicSample SamplePath="saints.mid" Pitch="7" Tempo="0.5" Volume="100" />
  </AMSCase>
  <AMSCase Name="Bored">
    <PADModel Pleasure="-1" Arousal="-1" Dominance="-1" />
    <MusicSample SamplePath="saints.mid" Pitch="-3" Tempo="2" Volume="50" />
  </AMSCase>
  <AMSCase Name="Neutral">
    <PADModel Pleasure="0" Arousal="0" Dominance="0" />
    <MusicSample SamplePath="saints.mid" Pitch="0" Tempo="1" Volume="64" />
  </AMSCase>
</AMSRepository>
```

Table 1. XML representation of AMS repository

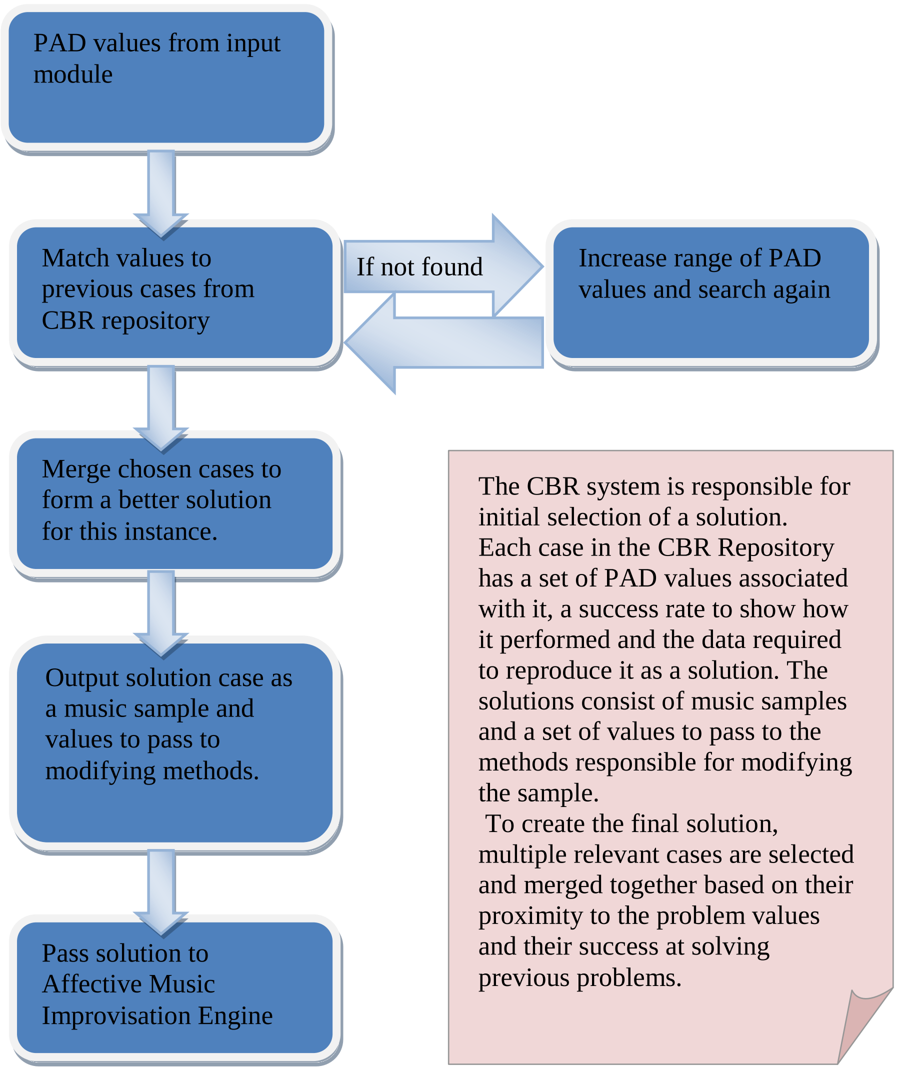

Figure 3. CBR System Flow Diagram

4.2.2 Affective Music Improvisation Engine (AMIE)

AMIE performs batch mode affectivisation of music using the Toub.Sound.Midi library. *alterSequence*(...) requires a MusicSample and three values to be passed in which reflect how to alter the pitch, tempo and velocity (volume) of the music sample. Once altered, the music sample is returned to the caller. *alterPitch*(...) alters only the pitch of a passed in MusicSample. *alterTempo*(...) alters only the tempo of a passed in MusicSample. *alterVelocity*(...) alters only the note velocity of a passed in MusicSample.

The pitch alteration algorithm uses a method from the external library to alter the pitch. The tempo alteration algorithm requires the discovery and alteration of a single attribute of the given sample. Finally, the velocity alteration algorithm loops through each note in the music sample setting a new note velocity. A major improvement to AMIE could involve the ability to affectivise musical pieces in an improved manner by employing MPEG-SMR for a comprehensive affectivisation parameter set.

Other improvements could involve addition of new effects such as using different "backing tracks" or "flourishes" into the sample to alter the emotional context of the music.

4.2.3 Analysis Engine

The Analysis Engine analyses the success of a solution by using the specified PADModel to alter the solution case. The altered case is then returned to the caller for inclusion into the CBR repository. The analysing algorithm modifies the original case so that its PADModel is equal to the feedback. Improvements to be made here will include accounting for other times this case has been supplied, as well as possible modifications for the cases used to generate this case.

5 REAL-TIME MUSIC AFFECTIVISATION (RMA)

The Affective Music Synthesis consists of a sub-system (developed in Pure data – Pd) with a graphical user interface based on GrIPD) known as Real-time Music Affectivisation (RMA). This sub-system takes as input a MIDI sequence and a stream of PAD values to output an affectivised version of the original MIDI sequence that reacts on-the-fly to varying emotive input.

Upon a change in the user's mood i.e. the emotive input (represented by PAD) is communicated to the Affective Music Synthesis server which computes corresponding affectivisation parameters (pitch, tempo and velocity) on-the-fly using a case-based repository containing the octants of temperament space mapped to the affectivisation parameters of the music piece. Each time the server receives data on the emotive state of the user; it retrieves the corresponding affectivisation parameters from the repository, if exactly matched; else it computes the parameters by looking at the closest two matches. The retrieved/computed parametric data is then sent back to the real-time sub-system which modifies the sequence accordingly.

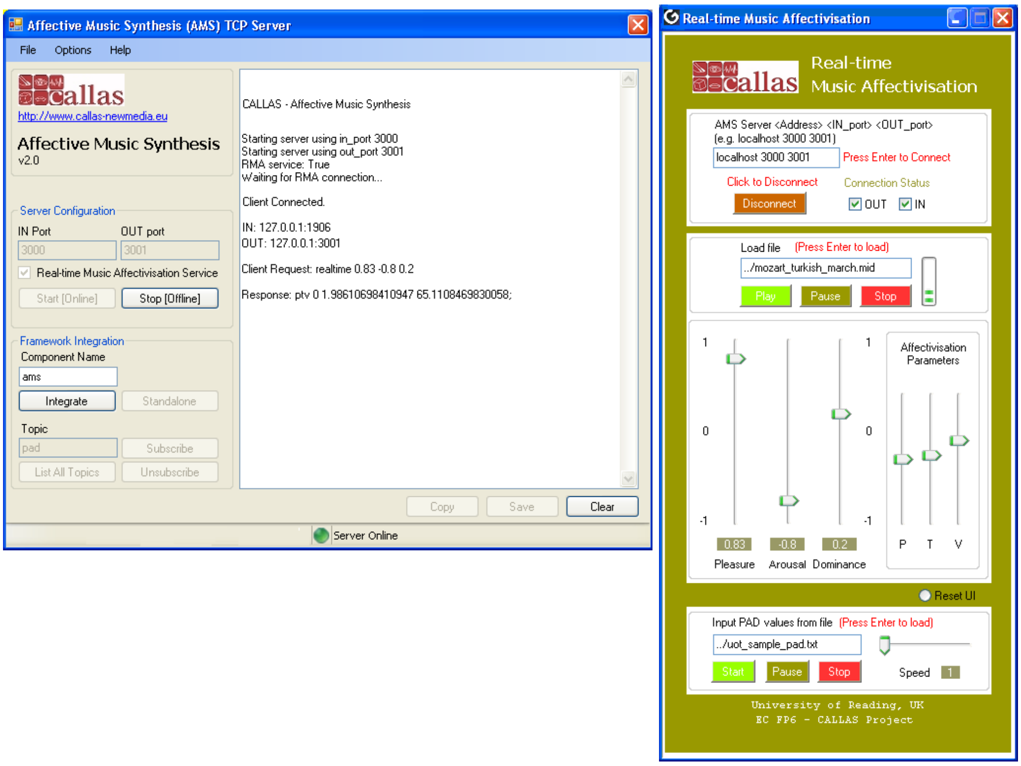

Figure 4. AMS Server and RMA GUIs

6 TCP BASED INTEGRATION: AMS SERVER AND RMA

The Real-time Music Affectivisation (RMA) patch connects as a TCP client to the AMS component's TCP server and uses a number of services using several commands. The RMA client initiates the data transfer from server by sending the init command. The server in turn starts to send PAD data along with CBR-matched-and-retrieved corresponding affect parametric data to RMA in real-time (assuming the framework has relevant data available from shelf-input components). The RMA client includes real-time music affectivisation logic, which would use these values for music synthesis and affectivisation.

It is possible to either start the TCP server of AMS directly, or provide command line arguments specifying ports for TCP for communication with RMA and UDP for OSC communication with the framework. Once connected, the RMA can use the following commands to communicate with the server for a required functionality.

- getcase <int>: Returns the case at the specified index in the repository.
- savecase <xml>: Saves the case to the repository. If the <xml> contains a valid index, then the case will replace any found at that index in the repository.
- delcase <int>: Deletes the case found at the specified index in the repository.
- generate <xml>: Generates a case based on the <xml> of an empty case with relevant PAD values. loadrepos <xml>: Loads the <xml> representation of the repository into the server.
- Saverepos: Retrieve the <xml> representation of the repository.
- Exit: Closes the connection with the server.

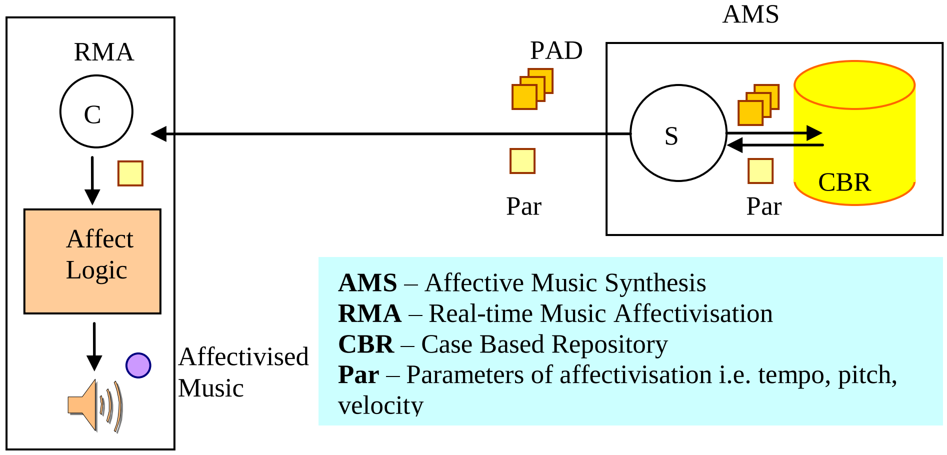

Figure 5. TCP communication between RMA and AMS

7 OSC BASED INTEGRATION: AMS AND CALLAS FRAMEWORK

AMS component communicates with the CALLAS Framework using OSC messages over a UDP socket. This is done in order to receive an integrated emotive state from various live shelf-input components of CALLAS from the framework.

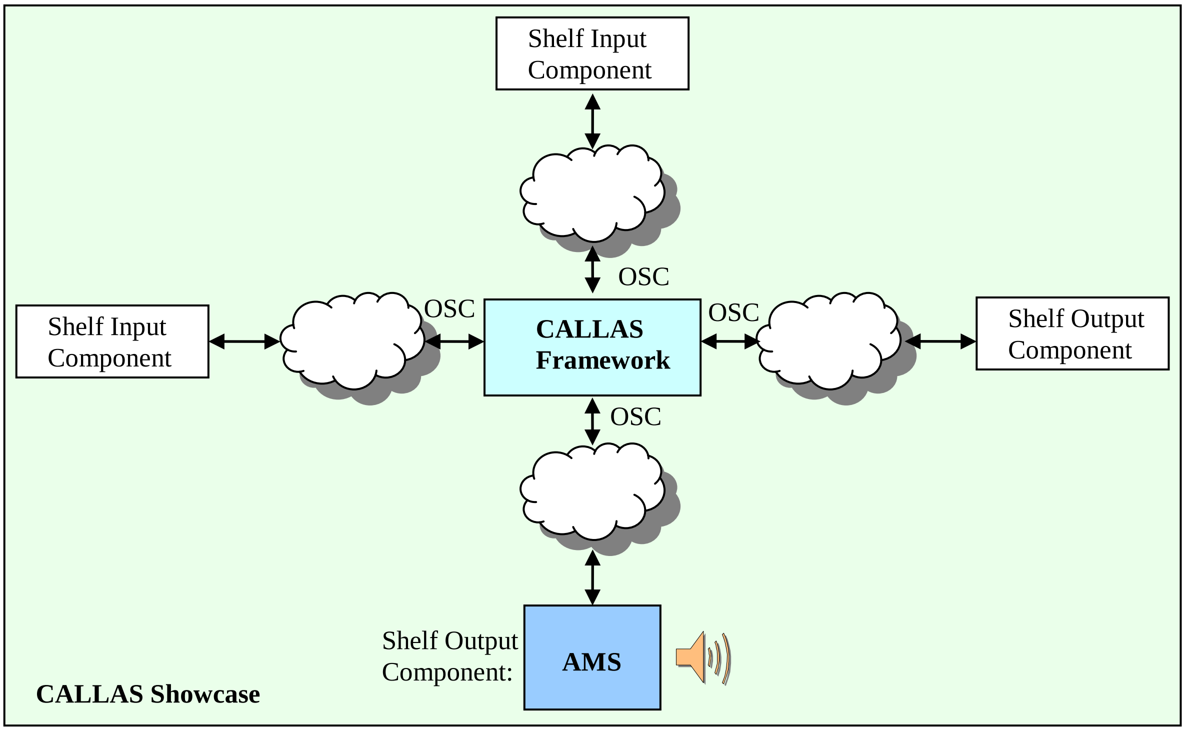

Figure 6. OSC Integration of AMS with CALLAS Framework

8 AFFECTIVE MUSIC SYNTHESIS IMPROVISER

We are currently in the process of developing an Affective Music Synthesis Improviser to synthesise percussive rhythms using a repository of recorded samples each symbolising a percussion note. Using a look-up table, this system would determine the various music characteristics to suit a particular user mood. Such a system would involve compiling together various pre-recorded samples together to form a short loopable piece of music.

This piece could then also be streamed to the real-time subsystem for real-time affectivisation based on the emotive input that drove the sample selection process. Together with the improviser add-on and the real-time music affectivisation sub-system, AMS would be able to remove, add and/or update new patterns on to the synthesised music as the user's mood changes from one state to another. For example, if a mood change requires the tempo to be increased (upbeat), the real-time sub-system could increase the tempo and the improviser add-on could update the beat pattern being used resulting in an enhanced affectivised output; conversely the improviser could react with a downbeat rendition sympathetically.

9 ONGOING AND FUTURE WORK

9.1 MPEG-SMR

Future directions for our system includes incorporation of the MPEG Standard for Symbolic Music Representation also known as MPEG-SMR to enhance the parameters of affectivisation of music (currently three parameters provided by MIDI are used: pitch, velocity and tempo). This would also open avenues for making use of music score in the affectivisation process.

9.2 Variable Mapping of PAD Model to Affectivisation Parameters

Another interesting development on the agenda involves the addition of a feature in our component to use alternating mappings of emotional models with affectivisation parameters. The system may then be setup for emotive input to be mapped in a directly or an inversely proportional fashion to the affective output, hence enabling the system to either resonate with or counter the user's mood. To make things more interesting, an automatic threshold switch for alternating between the two mappings could also be used. This threshold point could either be statically or dynamically computed by maintaining a history of user moods and affectivisation services rendered to use in a pre-specified time window.

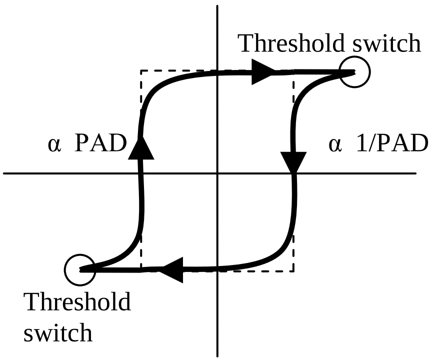

Figure 7. Hysteresis loop for two mappings of PAD Model with Affectivisation Parameters

10 CONCLUSION

The core components of AMS; an AMS prototype performing batch affective processing using toub.midi library; and a real-time music AMS prototype (RMA) using Pure data; have already been developed. Furthermore, Native and OSC integration of AMS with the CALLAS Framework have been successfully carried out. Future work will build on the achievements to-date to develop:

- Two types of emotive-state-to-affective-output mappings (directly- and inversely-proportional) including a manual or an automatic trigger/switch
- An Affective Music Synthesis Improviser to compile affective music from a repository of pre-recorded short loopable samples
- Optimise improvisation in terms of design and implementation in order to allow it to better adapt to new music genres, cultures and affects.

Once complete, this work is intended to be used in new media showcases in the arts and entertainment domain e.g. an interactive installation space providing dynamic musical/percussive real-time response to users/visitors in respect to the mood as captured by the various CALLAS Shelf components for affective input acquisition.

References

Mehrabian, A. 1987. 'Eating characteristics and temperament: General measures and interrelationships'. *Springer-Verlag.* New York.

Mehrabian, A. 1991. Outline of a general emotion-based theory of temperament. In J. Strelau and A. Angleitner (Eds.), *Explorations in temperament: International perspectives on theory and measurement* (pp. 75-86). Plenum Press, New York.

MPEG Group on SMR Symbolic Music Representation. Available on WWW: <http://www.interactivemusicnetwork.org/mpeg-ahg/>

Mehrabian. 1996. 'Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament'. *Current Psycho*., 14(4):261-292.

David B. Leake. 'CBR in Context: The Present and Future'. Accessed 23/04/2007. Available on WWW: <http://www.cs.indiana.edu/~leake/papers/p-96-01_dir.html/paper.html>

Christopher K. Riesbeck. *Inside Case Based Reasoning.* Lawrence Erlbaum Associates Inc.

Midi for .NET, *GotDotNet*, accessed 31/08/2007 Available on WWW: <http://www.gotdotnet.com/community/usersamples/details.aspx?sampleguid=89cde290-5580-40bf-90d2-5754b2e8137c>

Mehrabian, A. 'The PAD Comprehensive Emotion (Affect, Feeling) Tests'. Accessed 24/06/2007 Available on WWW: <http://www.kaaj.com/psych/scales/emotion.html>

Mehrabian, A. 'Personality Tests: A General Description of Temperament'. Accessed 24/06/2007, Available on WWW: <http://www.kaaj.com/psych/scales/temp.html>

Arcos, J.L., de Mántaras, R.L., Serra, X. 1998. 'SaxEx: a case based reasoning system for generating expressive musical performances'

NOOS, *IIIA Spain*, accessed 12/07/2007. Available on WWW: <http://www.iiia.csic.es/Projects/NOOS.html>

Milliman, R.E. 1982. 'Using Background Music to Affect the Behavior of Supermarket Shoppers'. *Journal of Marketing*, 46(3), 86-91. Available on WWW: <http://links.jstor.org/sici?sici=0022-2429(198222)46%3A3%3C86%3AUBMTAT%3E2.0.CO%3B2-7>

Grachten, M. 2001. ‚JIG: Jazz Improvisation Generator'. *In Proceedings of the MOSART Workshop on Current Research Directions in Computer Music, 1-6.* Barcelona, Spain: Pompeu Fabra University Publishers

Pereira, F. C., Grilo, C., Macedo, L., Cardoso, A. 1997. 'Composing Music with Case Based Reasoning (SICOM)'. *In Proceedings of Computational Models of Creative Cognition*. Available on WWW: <http://www.cisuc.uc.pt/view_pub.php?id_p=67>

Picard, R.W., Vyzas, E., Healey, J. 2001. 'Toward Machine Emotional Intelligence: Analysis of Affective Physiological State'. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. 23(10)

Chung, J., Vercoe, G.S. 2006. 'The affective remixer: personalised music arranging' *CHI Extended Abstracts*. 393-398

CALLAS Framework. Available on WWW: <http://www.callas-newmedia.eu/the-framework>

CALLAS Shelf. Available on WWW: <http://www.callas-newmedia.eu/the-shelf>

CALLAS Showcases. Available on WWW: <http://www.callas-newmedia.eu/the-showcases>

Pd – Pure Data. Available on WWW: <http://puredata.info/>

Max/MSP – Cycling 74. Available on WWW: <http://www.cycling74.com/>

GrIPD – Graphical Interface for Pure Data. Available on WWW: http://crca.ucsd.edu/~jsarlo/gripd/