Musical Composition by Autonomous Robots: A Case Study with AIBO

Eduardo R Miranda and Vadim Tikhanoff
Computer Music Research
Faculty of Technology, University of Plymouth
Plymouth, Devon PL4 8AA
United Kingdom
eduardo.miranda@plymouth.ac.uk, vadim.tikhanoff@postgrad.plymouth.ac.uk

Abstract

This paper presents a project whereby the Sony AIBO robot (ERS-220 model) was programmed to compose original music in real-time through direct interaction with its environment and people. It begins with a brief introduction to robotic applications in the arts and music. Then it introduces the technical issues encountered in realising the project (namely, AIBO's functionality and musical intelligence) followed by the proposed solutions. It presents *AIBO-Max,* a new tool that we implemented for programming AIBO combining Max/MSP and R-Code, and the musical composition system that is used in association with AIBO behaviours to generate the music.

1 Introduction

The field of robotics is becoming increasingly important to a number of research areas. Researchers in the fields of computer science, biology, and artificial life have found robotics to be an ideal platform with which to study intelligence, artificial evolution and various levels of communication; e.g. (Fugita and Kitano, 1998; Breazeal *et al.*, 2003, Greenman *et al.*, 2003; Iglesias *et al.*, 2004; Fukuda *et al.*, 2004). Moreover, the advancements of small, powerful and inexpensive microprocessors have had a catalytic effect on the emergence of new robotic applications and non-orthodox approaches to research in robotics, such as the work present in this paper: a robotics project combining musical composition and interaction.

Music and robotics may not be an association automatically made by the robotics community, but we are interested in exploring *biomorphic robotics* in the realm of *interactive music systems*. By biomorphic robotics we mean robots whose form resembles the form of biological living beings (e.g., dog-shaped robots or humanoids).

Interactive music systems are systems whose behaviour changes in response to some form of input, allowing them to be controlled in live performances of both notated and improvised music (Rowe, 1993; 2001). We are interested in designing autonomous robots that compose music interactively.

2 Robotics in the Arts and Music

The last twenty years has seen the accelerated exploration of robotics and machine-orientated movement by artists of many different backgrounds and disciplines; see (Wilson, 2001) for a review. The presence of robots can therefore be found in various realms of the artistic world, such as: theatre and dance, autonomy, extreme performance, destruction and mayhem**.** The artists' ever increasing interest in robotics is becoming more and more evident with the generation of shows and conferences around the world, from *Robotronika* (Website 1, 2005) in Vienna to Japan's ICC (InterCommunication Centre) show *Evolving with Robots* (Website 2, 2005). These events tend to look toward the artistic capabilities of modern robots while trying to break away from the notion of robots as passive slaves of "obligatory servitude" (Website 3, 2005). Interesting examples of artists using robots this way include Simon Penny (Website 4, 2005), Nicolas Anatol Baginsky (Website 5, 2005), Ken Rinaldo (Website 6, 2005) and Eduardo Kac (Website 7, 2005), to cite but four. The latter also compiled a list on robotic arts projects from 1995 to 2001 (Website 8, 2005).

Researchers and artists are constantly looking for new ways and means to develop robots in both the physical and cognitive sense. On the physical side, fluidity of movement and stability are paramount whilst the cognitive research focuses on the enhancement of the robots' interactive, comprehension and planning skills.

Various robotic forms and behavioural patterns are used in research, some resembling a human while others are modelled on animals. In some cases completely new kinds of behaviour are developed. Examples of artists working with robotic arts inspired by human behaviour are Eduardo Kac and Marcel.li Antunez Roca (Kac and Roca, 1997).

Although there have been some initiatives on using robotics to compose music interactively (Wasserman *et al.*, 2000; Suzuki *et al.*, 2000; Singer *et al.*, 2005; Birchfield *et al.*, 2005), we believe that the potential benefits of combining autonomous and biomorphic robotics and interactive music waits for further investigation. Our work takes *robot-human interaction* and uses it to create original music. Note, however, that we use the term "robot-human" instead of "human-robot" interaction because we wish to emphasise that we are interested in designing autonomous robots that compose music by interacting in the environment and with people, rather than systems where humans compose music by interacting with a robot.

3 Technical Issues

One of the main advantages we have, in relation to the development of this project, is that the interaction of human and biomorphic robots is easy and palatable (Kaplan, 2001).

We have chosen AIBO, due to the fact that AIBO is a very popular, affordable biomorphic robot. AIBO is also ideal for such a research project as it is fully programmable. Also, due to the fact that AIBO is a commercial robot, Sony has made sure it looks friendly in order to appeal to a big audience, particularly children (Kaplan *et al*., 2002).

In order to realize this project we had to solve two major technical problems:

- AIBO's standard programs
- AIBO's musical intelligence

The functionality of AIBO as standard is somewhat limited for the purposes of our project due to the lack of suitable programming tools for the types of systems we are interested in developing. Since this robot is commercialised as a toy or pet robot, the company favours the provision of programming tools oriented for designing applications for their target market.

Another problem is to furnish AIBO with musical intelligence. By musical intelligence we mean the ability to "compose" original music rather than merely reproduce a sound in response to input stimuli.

The AIBO community created a number of "behaviours" and interesting "personalities" (i.e., short programs that execute an action is response to certain stimuli). There are also competitions available for AIBO's most original behaviour or personality. Although there is a great deal of interest in these, they are mainly for actions with AIBO's actuators, without interacting with any external environment and most definitely not for music. Another serious drawback is AIBO's audio capabilities: they are of very poor quality. We needed to enhance AIBO's audio abilities to further implement AIBO's musical intelligence.

4 AIBO-Max: A New Programming Tool

The AIBO is fully programmable and this can be achieved at various levels, making it ideal for research and education. As well as being fully programmable the AIBO also offers researchers a stable development platform. However, the majority of AIBO owners would be using the Sony special R-Code language to programme their robotic pets and by doing so teach them new tricks and behaviours.

The AIBO robot was obviously not designed to create music. In order to alleviate this problem we developed *AIBO-Max*, a new programming tool for AIBO combining Max/MSP, Jitter and R-Code. Max/MSP and Jitter are well known commercial programming tools used by musicians and artists (Website 9, 2005) and R-Code is one of Sony's open standard for robotics development (Website 10, 2005).

The AIBO Software Development Environment allows AIBO to be controlled in two different ways, consisting of software that executes on AIBO itself or software which runs on a PC, which in turn connects to AIBO using a wireless LAN connection. Within the Software Development Environment there exists three varieties: Open-R SDK, R-Code SDK, and AIBO Remote Framework (Website 11 2005).

We needed to develop AIBO-Max because of the fact that the AIBO SDK is lacking in the kinds of music processing tools that we are interested in for this project. Max/MSP and Jitter, however *do* provide such tools (e.g., generative music facilities, pattern and advanced colour recognition). In addition, the music community has used them for over fifteen years.

Developed in the late 1980's at IRCAM in Paris, Max/MSP has become a fundamental part of new trends in interactive computer music and due to its capacity for real-time live performances it has also become of interest to sound and visual artists alike (Winkler, 2001). The main features of Max/MSP include:

- Support for unlimited MIDI input output streams (MIDI is a standard communications protocol used in music technology (White 2000))
- Interactive debugging and program editing features
- A cross-platform SDK which allows easy upkeep of a large community of users

- An object collection covering all basics of sampling, synthesis, and signal processing
- MME, DirectSound, and ASIO audio hardware support on windows
- Graphical filter design and envelope/function generator interfaces
- Support for building polyphonic MIDI-controlled synthesizers and samplers
- Hosts VST plug-ins and synthesizers

A second major development by way of add-ons came in 2003 with the release of Jitter (Website 12, 2005). Jitter is an extension, created for Max/MSP, which is essentially comprised of 135 video, matrix and 3-D objects. The presence of the Jitter objects enables the functionality of Max/MSP to be extended in order to generate and manipulate matrix data. In other words any data, which can be represented by rows and columns including, still images or film to spreadsheet data. Jitter is also useful to those with an interest in real-time processing, audio- visual interaction, data visualization and analysis.

Max/MSP and Jitter consist of graphical programming tools, which can be implemented in a variety of subject areas from computer-based music to image processing and analysis.

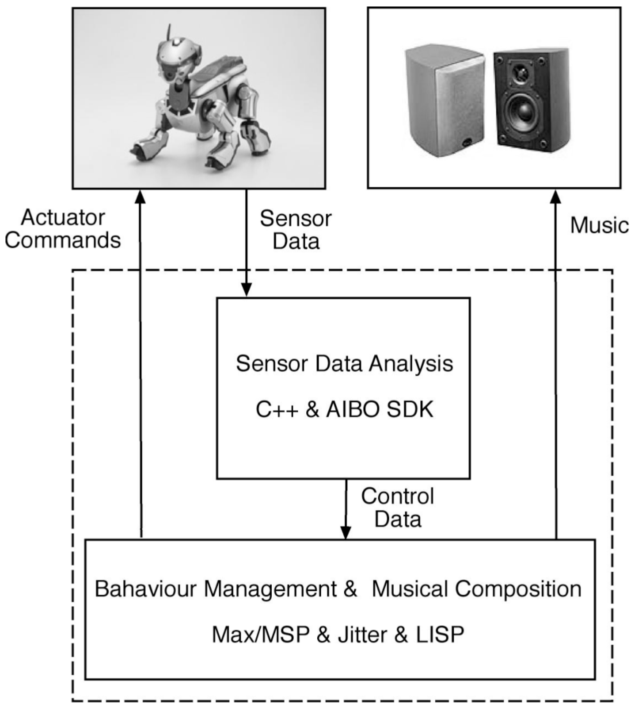

Figure 1: Block diagram of AIBO-Max.

Figure 1 shows a block diagram of AIBO-Max. Through the Max/MSP interface one is able to start the background C++ application, which in turn connects to AIBO and initiates the transfer of data. This transfer includes the images captured by AIBO and the data taken from each of its sensors. Once the connection to AIBO is made, the C++ application will decode the data coming from the wireless connection between AIBO and the PC and report through the Max/MSP interface whatever is happening or triggering the sensors. This is done by sending MIDI messages from the C++ application to Max/MSP. In turn, Max/MSP can therefore analyse each signal (triggered by a sensor) accordingly. These methods give Max/MSP more power to analyse data such as the vision system, comprising of pattern recognition and colour recognition. In return, Max/MSP allows for rapid prototyping of software to control all main aspects of AIBO (LEDs, switches and motors) and the music composition system.

The images sent from AIBO's camera are transmitted to Jitter which makes it possible to analyze the images and start the pattern recognition detecting shapes and coloured objects. Once a colour or object has been recognised and identified, Jitter will then send the C++ application data which will be transmitted to AIBO and actions will be taken accordingly; this information is also used to steer the music composition system. All these processes are in real time.

5 The Music Composition System

Broadly speaking, systems that generate music automatically can be classified as *abstract algorithmic models* or *music knowledge models*. Abstract algorithmic models generate music from the behaviour of algorithms that were not necessarily designed for music in the first instance, but embody pattern generation features that are suitable for producing musical materials. Such algorithms include Chaos (Little, 1993), Cellular Automata (Hunt *et al.*, 1991; Miranda, 2002) and Particle Swarms (Blackwell & Bentley, 2002) to cite but three examples. Music knowledge models generate music using algorithms derived from or inspired by well-established music theory. These systems often use Artificial Intelligence techniques and most of them can learn compositional procedures and rules from given examples. Historically, the latter have adopted either a symbolic approach (Steedman, 1984; Cope, 2001) or a connectionist (neural networks) approach (Mozer, 1994; Kohonen *et al.*, 1991), depending on the way they store information about music. Hybrid systems also exist (Biles *et al.*,1996).

While abstract algorithmic models have been successfully used to compose music, we propose that music knowledge models are better suited for the design of robotic intelligent systems that can handle musical concepts in meaningful ways (Wiggins and Smaill, 2000). These systems are often based on formalisms such as transition networks or Markov Chains to re-create the transition-logic of what-follows-what, either at the level of notes (Kohonen *et al.*, 1991) or at the level of similar "vertical slices" of music (Cope, 1996; 2001). For example, David Cope uses such example-based musical-generation methods but adds phrase-structure rules and higher-level composition structure rules (Cope, 2001). The act of combining the building blocks of music material together with some typical patterns and structural methods has proved to have great musical potential. This type of self-learning predictors of musical elements based on previous musical elements could be used on any level or for any type of musical element such as: musical note, chord, bar, phrase, section, and so on. However, there must be logical relations on all those levels; if a musical note is very close related to its predecessor(s) then a list of predecessors can predict quite well what note will follow. The same holds true for chords. It is more difficult to define the characteristics for phrase- and section-level elements, but it is not impossible.

We implemented a statistical predictor for AIBO at the level of small vertical slices of music such as a bar or half-bar, where the predictive characteristics are determined by the chord (harmonic set of pitches, or pitch-class) and by the first melodic note following the melodic notes in those vertical slices of music (see example below). We added a simple method of generating short musical phrases with a beginning and an end that also allows for the real-time influence from AIBO behaviours; the connection between these behaviours and the music will become clearer in the next section.

The system generates phrases of music by defining top-level structures of sentences and methods of generating similarity- or contrast-relationships between phrases. These look like this (LISP-like notation):

```
S → (INC BAR BAR BAR BAR BAR
     HALF-CADENCE 8BAR-COPY)
```

From this top-level, we then generate rules for selecting a valid musical building block for each symbol, including rules for incorporating AIBO behaviours in all decisions. For example:

```
INC → ((EQUAL 'MEASURE 1)
       (EQUAL 'STYLE-SET AIBO-BEHAVIOUR))

BAR → ((CLOSE 'PITCH 'PREV-PITCH-LEADING)
       (CLOSE 'PITCH-CLASS
              'PREV-PITCH-CLASS-LEADING)
       (EQUAL 'STYLE-SET AIBO-BEHAVIOUR))
```

This already defines a network that generates a valid sentence with a beginning and an end, including AIBO behaviour control through the variable `AIBO-BEHAVIOUR`. The generative engine will find a musical element for each of the constraint-sets that are generated above from `INC` and `BAR`, by applying the list of constraints in left-to-right order to the set of all musical elements until there are no constraints left, or there is only one musical element left. This means that it can happen that some of the given constraints are not applied. We illustrate this selection process below with an example-database.

The database of musical elements contains knowledge on various musical styles, with elements tagged by their musical function such as *measure 1* for the start of a phrase, *cadence* for the end, *style-set* for the style of the music material, and the special tags *pitch* and *pitch-class* that are both used for correct melodic and harmonic progression or direction. Table 2 shows an example database containing elements of only one music style: jazz.

ID	CO	P-CLASS	P	PCL	PL	TPE
JAZZ-1-1-CAD	JAZZ	((0 2 7)(0 2 4 5 7 11))	74	((0 4 9)(0 2 4 5 7 9 11))	76	CAD
JAZZ-1-1-MEA-6	JAZZ	((5 9)(0 5 7 9))	81	((0 2 7)(0 2 4 5 7 11))	74	BAR
JAZZ-1-1-MEA-5	JAZZ	((0 4)(0 4 7))	76	((5 9)(0 5 7 9))	81	BAR
JAZZ-1-1-MEA-4	JAZZ	((0 4)(0 3 4 6 7 9))	83	((0 4)(0 4 7))	76	BAR
JAZZ-1-1-MEA-3	JAZZ	((0 4)(0 3 4 6 7 9))	76	((2 7 11)(2 5 7 9 11))	83	BAR
JAZZ-1-1-MEA-2	JAZZ	((2 7 11)(2 5 7 9 11))	83	((0 4)(0 3 4 6 7 9))	76	BAR
JAZZ-1-1-MEA-1	JAZZ	((0 4)(0 3 4 6 7 9))	76	((2 7 11)(2 5 7 9 11))	83	INC

Table 1: An excerpt from the database of musical elements. CO = style set, P-CLASS = pitch class, P = pitch, PCL = pitch-class leading, PL = pitch leading and TPE = type.

Table 1 shows the main attributes that are used to recombine musical elements. `P-CLASS` (for *pitch-class*) is a list of two elements. The first is the list of start-notes, transposed to the range of 0-11. The second is the list of all notes in this element (also transposed to 0-11). `P` is the *pitch* of the first (and highest) melodic note in this element; by matching this with the melodic note that the previous element was leading up to we can generate a melodic flow that adheres in some way to the logic of "where the melody wants to go". The `PCL` (for *pitch-class leading*) elements contain the same information about the original next bar; this is used to find a possible next bar in the recombination process. Then there are the `INC`, `BAR`, and `CAD` elements. These elements are used for establishing whether these elements can be used for phrase-starts (`incipient`), or cadence.

Simply by combining the musical elements with the constraint-based selection process that follows from the terminals of the above phrase-structure rewrite-rules, we end up with a generative method that can take into account real-time changes of AIBO behaviour. This generates musical phrases with a domino-game like building block connectivity:

```
((EQUAL 'MEASURE 1)
 (EQUAL 'STYLE-SET AIBO-BEHAVIOUR))
```

Assuming that there are also musical elements available from styles other than JAZZ, the first constraint will limit the options to all *incipient* measures from all musical elements from all styles. The second constrains will then limit the options according to the current behaviour to one style that is associated with the respective set of behaviours, as follows:

```
((CLOSE 'PITCH 'PREV-PITCH-LEADING)
 (CLOSE 'PITCH-CLASS
        'PREV-PITCH-CLASS-LEADING)
 (EQUAL 'STYLE-SET AIBO-BEHAVIOUR))
```

In the given phrase structure, the rule that follows from BAR then defines the constraints put upon a valid continuation of the music. These constrains will limit the available options one by one and will order them according to the defined by rule preferences. The CLOSE constraint will order the available options according to their closeness to the stored value. For example, after choosing:

```
(JAZZ-1-1-MEA-1
P-CLASS ((0 4) (0 3 4 6 7 9))
P 76
PCL ((2 7 11) (2 5 7 9 11))
PL 83
BAR INC
CO JAZZ)
```

At the beginning, PREV-PITCH-LEADING will have stored 83, and PREV-PITCH-CLASS-LEADING will have stored ((2 7 11) (2 5 7 9 11)). This will result in measure 2 and 4 being ranked highest according to both pitch and pitch-class, and measure 6 and the cadence close according to pitch-class, while measure 6 is also quite close according to pitch. This weighted choice will give a degree of freedom in the decision that is needed to generate pieces with an element of surprise. The music will not get stuck in repetitive loops, but it will find the closest possible continuation when no perfect match is available. AIBO can still find a close match in this way if the third constraint eliminated all the obvious choices that are available; e.g., because a jump is requested to the musical elements of another style, who might not use the same pitch-classes and pitches.

6 The Robotic Music System

AIBO has a variety of sensors located at various points around the frame of the robot. These include touch sensors, temperature sensors, tilt sensors, infra-red sensors and a functioning camera. For the purpose of this project the two most important sensors are the infra-red and the camera. The infra-red distance sensor allows AIBO to detect objects within its vicinity and avoid collision by navigating around the objects. The camera enables AIBO to "distinguish" and "recognize" colours and shapes in its environment. The infra-red and the camera together form a fundamental structure for the robot-human interaction.

The AIBO is set out in a predefined environment measuring a few square metres. Within this space there are a variety of coloured objects. AIBO navigates around this predefined space by means of obstacle and perimeter avoidance and interacts with objects resulting in the creation of music (Figure 2). By *interacting* with objects, we mean that AIBO "perceives", through the use of the camera, the various coloured objects and depending on the shape, size and colour of the objects a behaviour is initiated, which in turn affects the musical output.

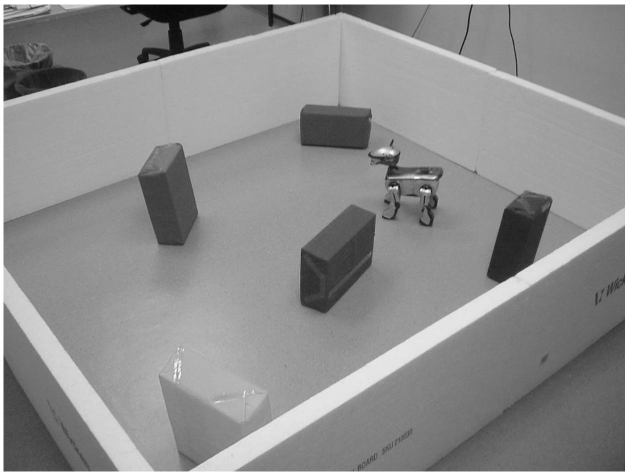

Figure 2: AIBO in its environment.

In technical terms, AIBO behaviours are short programs that execute an action. This notion of behaviour is inspired by research in emergent behaviour, which suggests that in nature animals (such as insects) with a very low level of "computational ability" (sic) can perform quite complex autonomous tasks (Jensen *et al.*, 1998). These natural behaviours pave the way for research into artificial intelligence and the creation of robots that can perform such complex autonomous tasks (Nehmzow, 1999).

The word "behaviour" in the context of this research relates to the way in which AIBO "perceives" its surroundings and in turn responds physically to what has been "perceived". This process gives the robot autonomy to deal with situations on a real time basis, as and when they occur.

It is possible for people to enter the predefined environment wearing coloured boots and gloves and interact with AIBO. Entering the space will catch the attention of AIBO, which in turn will stop its activity and make its way towards the person. AIBO will then stop in front of the person and start interacting with him/her which will then affect the behaviour and the composition.

In table 2 we can see examples of AIBO behaviours, which are associated with generative musical processes (as explained in the previous section) that compose music on the fly. The behaviours can influence in a well-defined way the mixture of different style-elements programmed in the system. It can generate music that contains, for example, different musical elements in response to one set of behaviours as to another set of behaviours. These associations between behaviours and style-elements are setup beforehand by means of a simple user interface.

Reactive behaviours	**Musical Processes**
Obstacle avoidance	Various Musical Processes 1 (1a, 1b etc...)
Colour reactions	Various Musical Processes 2 (2a, 2b etc...)
Human presence reaction	Various Musical Processes 3 (3a, 3b etc...)
Directional reaction	Various Musical Processes 4 (4a, 4b etc...)
Emotional behaviours	**Musical Processes**
"Happy behaviour"	Musical Process 5
"Sad behaviour"	Musical Process 7
"Dislike behaviour"	Musical Process 8
"Curiosity behaviour"	Musical Process 9
"Playful behaviour"	Musical Process 10

Table 2: Certain examples of musical processes associated with behaviours.

There are two categories of behaviour: *reactive* and *emotional* (Table 2). Within the reactive behaviour for example, when a human presence is detected, AIBO will direct itself towards that person, welcome them into the environment by sitting down in front of them, look up and greet them by waving its paw, displaying lights and making music (using musical processes of group 3). Similarly when AIBO detects an obstacle it will stop its course, take a few steps backwards, examine the surroundings and take another path, while generating music (using musical processes of group 1).

Within the emotional behaviour, for example when AIBO will display a "happy" behaviour it will generate music accordingly (musical process 5) and perform a dancing-like behaviour while flashing lights.

The objects in AIBO's environment are of various size, shape and colour. The most important of these features is colour as AIBO essentially tracks the colour of the objects, which results in changes of behaviour and music composition. The importance of colour is reflected in the robot-human interaction, as a person entering the environment would wear coloured boots and gloves, allowing AIBO to track the person. The majority of the interaction is drawn from the movements and hand gestures made by the person as this will have a direct effect on AIBO's behaviour.

The musical aspect here is extremely individual as the AIBO robot is creating and composing original music, rather than merely reproducing pre-programmed loops, or pre-recorded sounds. The technical and the artistic expression come together, as the music is produced by robot-human or robot-environment interaction.

7 Concluding Remarks

We have implemented a system whereby AIBO can create original music, compiled through a musical knowledge-base, governed by interaction with its surrounding environment. We are currently adapting our model to allow AIBO to analyse incoming music with a view on using music as an additional input to drive AIBO's behaviour. Also, we are adapting the music composition system so that the compositional rules are induced from given examples of music using classic grammar induction techniques; e.g. (Honavar *et al.* 1998). This in turn will enable AIBO to alter its compositional strategies as a direct result of the (musical) interactions.

Due to the fact that we are dealing with musical composition, an assessment of the musical results would therefore be necessary. Details such as the compositional makeup and links between music and behaviour are recorded and monitored in order to assess the general musical output of AIBO. A report on this assessment will be submitted for publication in due time.

During the testing phase, we noticed that the reaction time capabilities of AIBO were relatively slow. In general terms this means that interaction with AIBO must include deliberate concise movements in order to gain a response. Flighty movements would just not be recognized. Although this was not a great hindrance, it is an area that could be improved in the future and with such improvements more sophisticated robotic musical interaction would be achieved.

The overall outcome of this project has also been to successfully merge two relatively distant areas of interest: music and robotics. That is to say the effective combination of artistic and engineering practices, which, allows artists and musicians to be creative yet have access to robotic technology. The result of which is that artists and musicians can use the AIBO robot as a means of expression without having to have the technological expertise to program the robot completely from scratch.

Acknowledgement

The authors would like to thank Bram Boskamp for his contribution on the design and implementation of the music composition system.

References

Biles, A., Anderson, P. G. and Loggi, L. W. (1996). *Neural Network Fitness functions for a Musical IGA*. Technical Report, Rochester Institute of Technology, USA.

Birchfield, D., Lorig, D. and Phillips, K. (2005). “Sustainable: a dynamic, robotic, sound installation”, *Proc. 2005 International Conference on New Interfaces for Musical Expression (NIME 2005)*, Vancouver, Canada.

Blackwell, T. M. and Bentley, P. J. (2002). “Improvised Music with Swarms”, *Proc. 2002 Congress On Evolutionary Computation*, Honolulu, USA.

Breazeal, C., Brooks, A., Gray, J., Hancher, M., McBean, J., Stiehl, W. D. and Strickon, J. (2003). "Interactive Robot Theatre", *Communications of the ACM*, 46(7):76-85.

Cope, D. (1996). *Experiments in Musical Intelligence*. Madison, WI: A-R Editions Inc.

Cope, D. (2001). *Virtual Music*. *Cambridge*. MA: The MIT Press.

Fujita, M. and Kitano, H. (1998). “Development of an Autonomous Quadruped Robot for Robot Entertainment”, *Autonomous Robots*, 5:7-18.

Fukuda, T., Hasegawa, Y. and Kajima, H. (2004). “Intelligent Robots as Artificial Living Creatures”, *Artificial Life and Robotics*, 8(2):101-110.

Greenman, J., Holland, O., Kelly, I., Kendall. K., McFarland, D., Melhuish, C. (2003). “Towards Robot Autonomy in the Natural World: A Robot in Predator’s Clothing”, *Mechatronics*, 13(3):195-228.

Honavar, V, Slutzki,G. (Eds.) (1998). *Proc 4th International Colloquium on Grammatical Inference*, LNCS 1433. Berlin: Springer-Verlag.

Hunt, A, Orton, R. and Kirk, R. (1991). “Musical Applications for a Cellular Automata Music Workstation”, *Proc International Computer Music Conference (ICMC 91)*, Montreal, Canada.

Iglesias, R., Kyriacou, T., Nehmzow, U., Billings, S. (2004). “Task Identification and Characterisation in Mobile Robotics”, *Proc. of TAROS 2004*, Department of Computer Science, University of Essex, Report Number: CSM-415.

Jensen, H. J., Goddard, P. and Yeomans, J. (Eds.) (1998). *Self-Organized Critically: Emergent Complex Behaviour in Physical and Biological Systems*. Cambridge: Cambridge University Press.

Kac, E. and Roca, M. A. (1997). “Robotic Art”, *Leonardo Electronic Almanac*, 5(5).

Kaplan, F., Fujita, M. and Doi, T. (2002). “Dans les entrailles du chien”, *La Recherche*, 350:84-86.

Kaplan, F. (2001). “Artificial Attachment: Will a robot ever pass Ainsworth’s Strange Situation Test?”, *Proc. Humanoid 2001 – IEEE-RAS International Conference on Humanoid Robots*, Tokyo, Japan.

Kohonen, T., Laine P., Tiits, K and Torkkola, K (1991), “A Nonheuristic Automatic Composing Method”, P. Todd and D. G. Loy (Eds.), *Music and Connectionism*. Cambridge (MA): The MIT Press.

Little, D (1993) Composing with Chaos; Applications of New Science for Music. *Interface* (now *Journal of New Music Research*) 22(1):23-51.

Miranda, E. R. (2002). “Voices of artificial life: on making music with computer models of nature”, *Proc. International Computer Music Conference (ICMC2002)*, Göteborg, Sweden.

Mozer, M. (1994). “Neural network music composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing”, *Connection Science* 6:247-280.

Nehmzow, U. (1999). *Mobile Robots: A Practical Introduction*. London: Springer-Verlag.

Rowe, R (1993) Interactive Music Systems. Cambridge, MA: The MIT Press.

Rowe, R (2001) Machine Musicianship. Cambridge, MA: The MIT Press.

Singer, E., Feddersen, J. and Bowen, B. (2005). “A Large-Scale Networked Robotic Musical Instrument Installation”, *Proc. 2005 International Conference on New Interfaces for Musical Expression (NIME 2005)*, Vancouver, Canada.

Steedman, M. (1984). “A Generative Grammar for Jazz Chord Sequences”, *Music Perception* 2:52-77.

Suzuki, K., Tabe, K., Hashimoto, S. (2000). "A Mobile Robot Platform for Music and Dance Performance", *Proc. of 2000 International Computer Music Conference*, Berlin, Germany.

Wasserman, K., Blanchard, M., Bernardet, U., Manzolli, J. and Verschure, P. (2000). “Roboser – An Automous Interactive Musical Composition System”, *Proc. of the International Computer Music Conference*, Berlin, Germany.

White, P. (2000). *Basic MIDI*. London: Sanctuary Publishing.

Wiggins, G. and Smaill, A. (2000). “Musical Knowledge: what can Artificial Intelligence bring to the musician?”, In E R Miranda (Ed.), *Readings in Music and Artificial Intelligence*. Amsterdam: Harwood Academic Publishers.

Wilson, S. (2001). *Information Arts: Intersection of Art, Science, and Technology*. Cambridge, MA: The MIT Press.

Winkler, T. (2001). *Composing Interactive Music: Techniques and Ideas Using Max*. Cambridge, MA: The MIT Press.

Web References:

Website 1 (2005). http://robot.t0.or.at/ (Accessed on 26 April 2005).

Website 2 (2005). http://www.nticc.or.jp/Archive/1999/ROBOT/Works/babot_e.html (Accessed on 26 April 2005)

Website 3 (2005). http://capek.misto.cz/english/ (Accessed on 26 April 2005).

Website 4 (2005). http://www.ace.uci.edu/penny/ (Accessed 26 April 2005).

Website 5 (2005). http://www.baginsky.de/ (Accessed 26 April 2005).

Website 6 (2005). http://accad.osu.edu/~rinaldo/ (Accessed 26 April 2005).

Website 7 (2005). http://www.ekac.org/interactive.html (Accessed 26 April 2005).

Website 8 (2005). http://www.ekac.org/robotichronology.html (Accessed 26 April 2005).

Website 9 (2005). http://www.cycling74.com/index.html (Accessed on 26 April 2005).

Website 10 (2005). http://openr.aibo.com/openr/eng/no_perm/faq_rcode.php4 (Accessed on 26 April 2005).

Website 11 (2005). http://openr.aibo.com/openr/eng/no_perm/faq_aibosde.php4 (Accessed 26 April 2005).

Website 12 (2005). http://www.cycling74.com/products/jitter.html (Accessed 17th February 2005).