

Two Highly Integrated Real-Time Music and Graphics Performance Systems

Robert Rowe (1)

Eric L. Singer (2)

(1) Media Research Laboratory, New York University
robert.rowe@nyu.edu

(2) Media Research Laboratory, New York University
esinger@cat.nyu.edu

Abstract

We describe two systems coupling animated graphic displays with real-time music analysis. One of these systems is for stage performance involving an ensemble of six instrumentalists and live video projection. The other allows users to perform with members of an animated virtual jazz band. *A Flock of Words* is a collaboration of composer Robert Rowe and video artist/holographer Doris Vila, with programming and animation design by Eric Singer. The innovations of this application are real-time animation of graphic objects in response to analysis of musical input, coupled with the simultaneous presentation of video, lighting, large-scale holograms, and algorithmically generated computer music. The *Interactive Virtual Musicians* project presents animated jazz musicians that can improvise with users performing on instruments or interacting via the mouse, video input, sensors and speech recognition. Interactive Virtual Musicians are artificially intelligent, autonomous animated characters which create a graphical and musical performance in real-time.

1 A Flock of Words

A Flock of Words was premiered at New York University in the spring of 1995. The instrumental ensemble for the piece consists of violin, viola, cello, MIDI keyboard and two percussionists. Their performance is tracked in real time through the MIDI keyboard and a percussion sensor.

1.1 Setup

The elements of the technical setup of the piece are as follows: 3 Apple Macintosh computers, 3 video projectors, 3 robotic lighting elements, a MIDI-controlled light board, a laser, 2 large holograms and sound synthesis equipment. The general equipment arrangement is shown in Figure 1. Two of the computers are used to cue up and play back video clips and to generate animations in real time. One of these (video computer 1) sends an identical signal to two video projectors, each projecting onto large scale holograms on either side of the instrumental ensemble. The second video computer projects a different set of videos and animations onto a large screen at the back of the stage.

All of the software for *A Flock of Words* is written in C by the authors. Performance information from the ensemble is analyzed by a program running on the master computer and looking for musical attributes such as register, density and articulation. Software on the video computers is responsible for receiving control messages from the analysis software and sending video and animation to the display projectors. The analysis and video machines communicate through MIDI connections and a set of

MIDI messages which we define. The master computer also generates a stream of MIDI-encoded music which is sent to a synthesizer and effects processor to accompany the live ensemble and control lighting effects through a dedicated show controller. Video, animations and lights are projected onto the screen behind the ensemble as well as the holograms flanking the ensemble, changing their appearance as the piece progresses.

Displayed video consists of prerecorded video clips and real-time generated animation. The video clips are stored and played back from QuickTime movie files. The animation is based on Craig Reynolds' "Boids" algorithm from an implementation by Simon Fraser. To create the *Flock of Words* animation, a set consisting of 10 to 30 words is selected from the text of *Crowds and Power* by Elias Canetti and animated using the flocking algorithm. The center point of each word (or "Woid") is treated as the center of a Boid and animated under real-time control. Numerous parameters are used to change the flocking attributes and influence the look of the flock. These include the speed and acceleration of the Woids; their tendency to stay close to the center of the flock; to avoid each other; to follow a point on the screen; and to avoid the edges of the screen.

Figure 2 shows four characteristic Woid flocks in mid-flight. The flock in the lower left, for example, is moving toward a point of attraction centered near the bottom of the screen. The flock above shows the Woids in a pattern of greater dispersal due to a change of direction or increase in neighbor avoidance. Because the objects are words, they continually enter into new grammatical relationships with each other as they fly, sometimes aligning as in the original

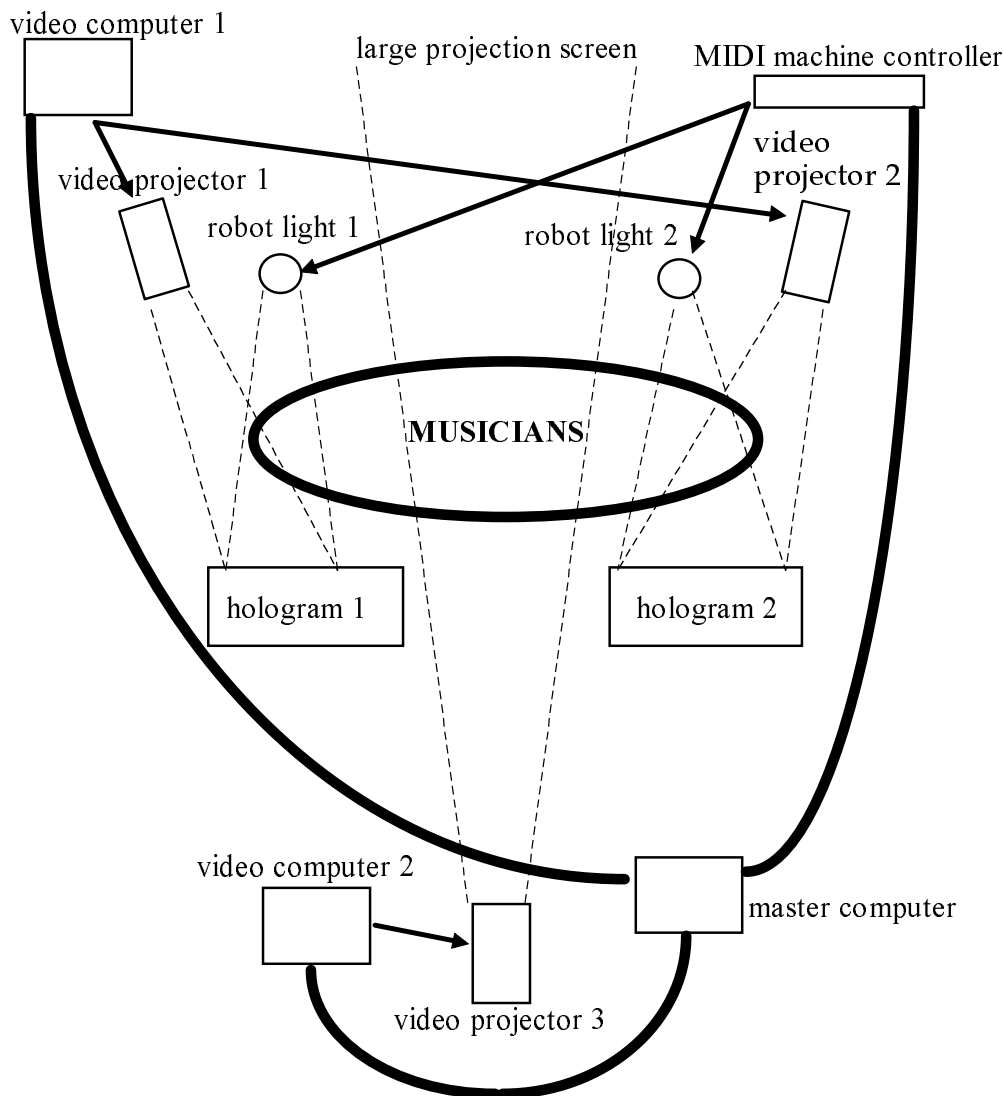


Figure 1: Stage setup for *A Flock of Words*

text, other times garbling the syntactic order entirely. The continual fluctuation between linear sense and nonsense was one of the motivations for this approach: the force of Canetti's original text is always present through the normal-order presentation in the video clips and balances with the non-linear presentation in the animation.

The analysis software sends MIDI messages to the video programs to control the displayed video. (MIDI communication between machines is indicated by the thick black lines in Figure 1.) We define various MIDI messages to be interpreted as control messages. For example, *note on* messages are used to initiate playback of QuickTime video, with the note number selecting the particular video clip to be played. Continuous-controller

messages are used to change the values of flocking parameters. Video control messages are sent by the analysis software based on cue points in the musical score as well as musical gestures and playing attributes of the live performers. In this manner, the displayed video is guided by the musical performance. By linking various flocking parameters to messages sent by the analysis software, which are in turn derived from musical gestures of the performers, we achieve the effect of the flock being controlled by the performers. The full set of video controls available to the analysis software includes selection of video clips; selection of word sets from the text; color of the Woids and background screen; and twelve flocking parameters.

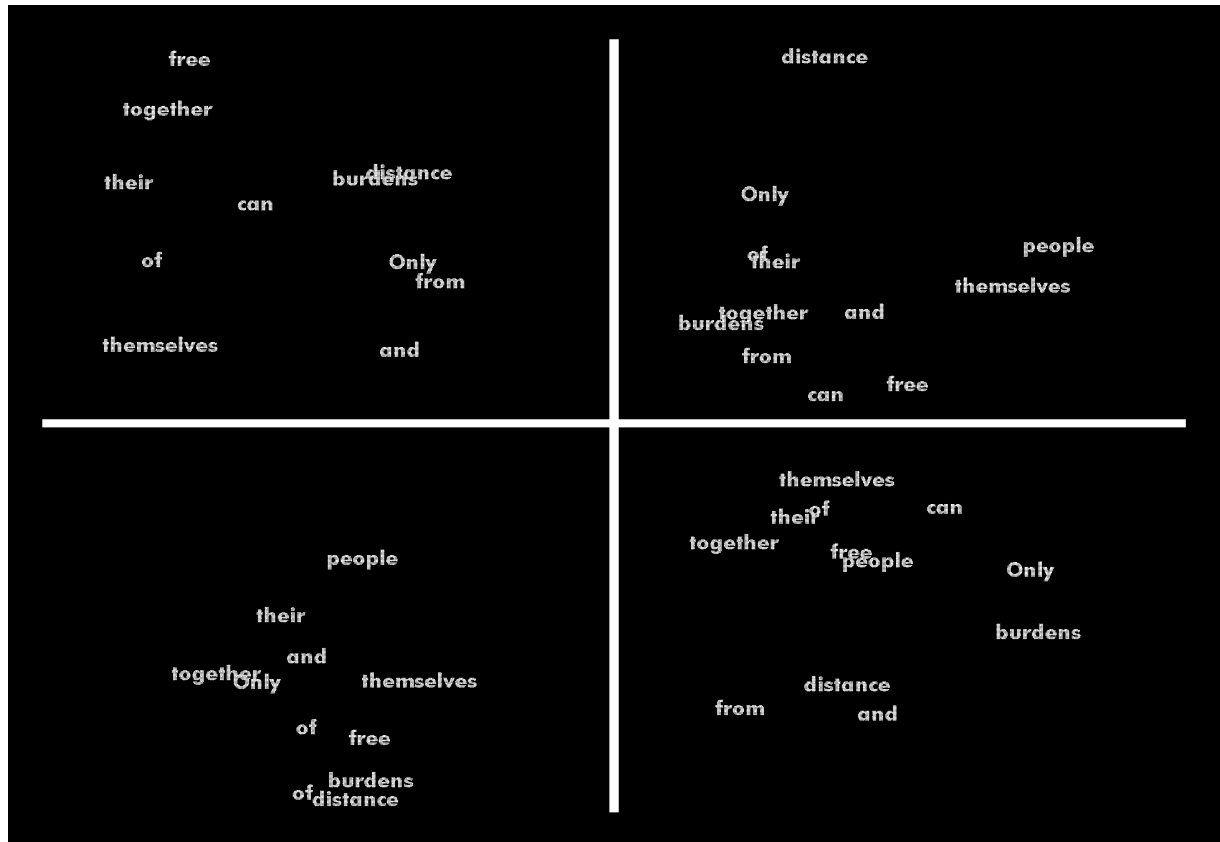


Figure 2: Screen displays of Woid flocks

1.2 Holography

Two large rainbow holograms (22”H x 42”W) were created by Doris Vila for the piece. Triple-exposure images on the hologram mix the spectral colors into bright fields filled with found objects, words and diagrams. Ordinarily, holograms are displayed by shining a point-source of light on the plate at the reconstruction angle. However, in *A Flock of Words*, video projectors illuminate the hologram. The holographic surface becomes a diffractive screen for the Woids animation. As the color of the Woids changes, it filters the color in the holographic reconstruction. In the piece’s second movement, robotic controls drive mirrors that alter the apparent location of the reconstruction light, making the holographic imagery cross and turn in response to the music.

1.3 Video

For the QuickTime video clips, we combined footage of flocking birds, animation of flying phrases and close-ups of hands and mouths pronouncing vowels. After-effects were applied to make sparks fly out of the mouths (Figure 3). For video playback, we

created the movies at half-size, and used compression to optimize colors and data rate. The movies are optimized for speed of playback to enable the visual rhythm to flow with the musical performance. During the performance, the video playback projects the movies at twice size, yielding full-screen images with smooth motion.

1.4 Installation Version

Another implementation of the project was developed for presentation at the Musik + Licht exhibition in Berlin in the fall of 1996. For this application, a screen is set before an active floor equipped with pressure sensors. The motion of the Woids across the screen corresponds to the position of viewers on the floor. Similarly, parameters of a compositional algorithm producing music are tied to the presentation of the Woids and controlled by the placement of the viewers. In this case, the analysis of musical performance is replaced by position mapping of participants in front of the screen. The analysis directs both the flight of the Woids and the production of the music.

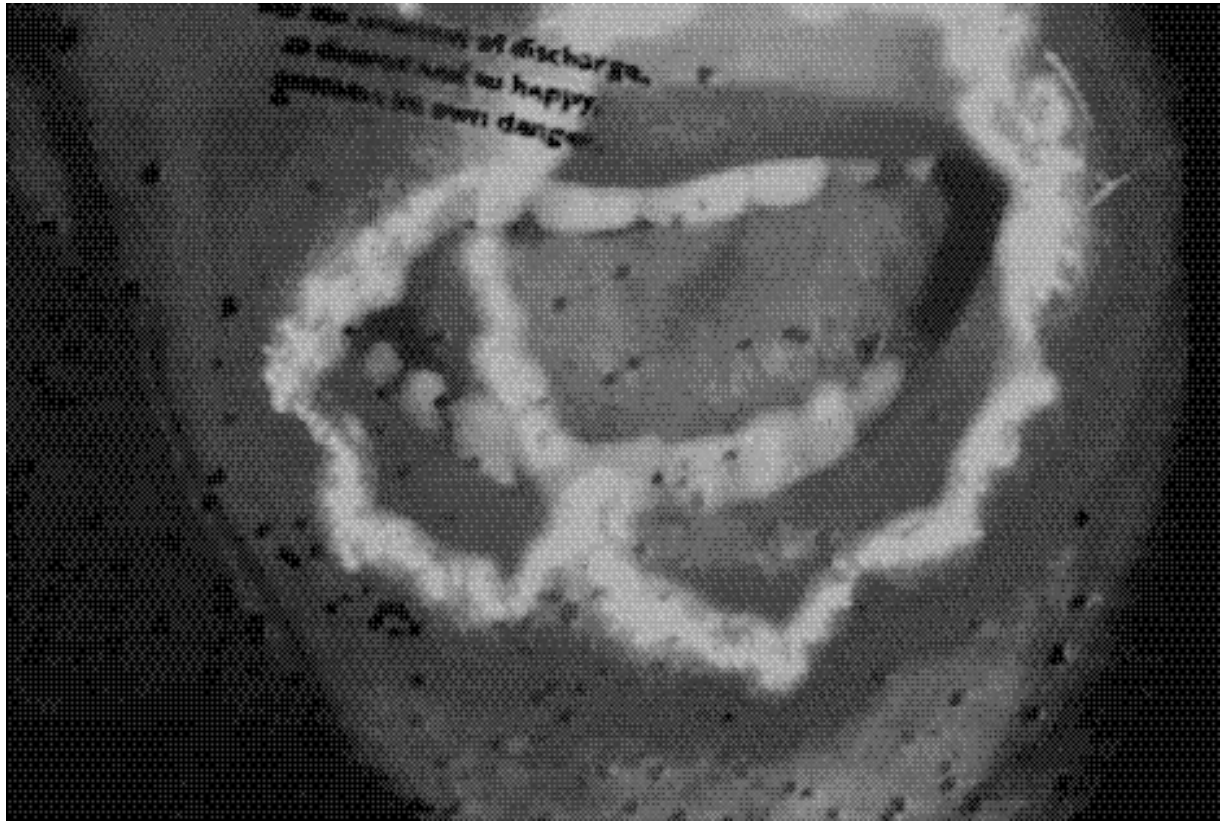


Figure 3: Still from *Flock of Words* video clip

2 Interactive Virtual Musicians

A second project, entitled Interactive Virtual Musicians, features computer-animated jazz musicians that can improvise along with a person performing on an instrument or interacting via a variety of input sources. Interactive Virtual Musicians builds upon a previous project featuring a user-conductible animated opera singer. We have extended this work to create more sophisticated characters capable of generating interactive graphical and musical performances in real-time.

2.1 Background: The Aria Project

At the SIGGRAPH '96 conference, we presented an installation entitled *Aria*, created in collaboration with the Laboratório de Sistemas Integráveis of University of São Paulo, Brazil. In this installation, a human participant approaches a video projection of an opera stage and picks up an electronic baton. Gigio, a computer-animated opera singer, takes the stage, bows and prepares to sing. As the participant begins to conduct, a MIDI orchestra plays, following the tempo and amplitude of the conductor's stroke. Gigio sings along using a singing synthesis algorithm and dramatizes the scene through actions, facial expressions and gestures which he chooses based on musical characteristics and score location.

Animation is realized in IMPROV. Developed by Ken Perlin et al. at the NYU Media Research Laboratory [Perlin 95] [Perlin & Goldberg 96], IMPROV is a system for creating animated characters with individual personalities. Singing synthesis is generated by MIT's Csound software using the FOF synthesis algorithm. The input and control for *Aria* is implemented in Opcode MAX. A MAX patch augmented with several custom external objects interprets conducting data from an electronic baton, drives MIDI sequences and Csound synthesis, and controls character animation. It communicates with IMPROV and Csound via Telnet connections.

Beats from the baton drive the sequence playback, controlling the tempo and amplitude of the MIDI orchestra and singing synthesis. Horizontal position controls the singer's vowel sound, who sings with an *a*, *e*, *i*, *o* or *u*. The character is sent messages indicating notes and score location. He synchronizes his facial expression to the vowel sound and chooses his actions based on the score location. For example, he knows when to make his entrance, where the climax of the music is and when to take his bow.

2.2 The IVM System

Following our success with *Aria*, we wished to create more advanced characters endowed with both musical and graphical

intelligence. We wanted them to interact with both musicians and non-musicians. Finally, we wanted to design a versatile system which could be configured for many different scenarios and installations. These goals led to the development of the Interactive Virtual Musicians (IVM) system.

The complete system consists of two major software subsystems: the IMPROV software and the IVM control software. IMPROV enables authors to create real-time behavior-based character animation. Characters in IMPROV, known as *virtual actors*, are artificially intelligent, autonomous and directable. They are endowed with a library of animated actions, movements and gestures as well as individual personalities created by a programmer or animator using a scripting system. Characters are then able to generate their own animation sequences based on external input and influences (such as user input, musical analysis software and the actions of other characters) and in accordance with their personality traits.

The IVM control software (which we refer to henceforth as IVM) is the main focus of our development work. It extends the functionality of the virtual actors to include music performance, thereby turning them into virtual musicians. IVM is responsible for receiving and interpreting the various forms of input, generating and playing the musical performance and directing the graphical performance of IMPROV.

Various modes of input are used to influence the musical accompaniment and improvisation performed by the virtual musicians as well as their animated performance. For example, a human musician can play chord changes while a virtual saxophonist improvises along, or a person can pick up the conductor's baton and lead a jazz trio. The mood of the performance is controlled by the user and is reflected by the virtual musicians in the tone of their musical performance and the body language of their animated performance. In this way, the IVM system presents an integrated visual and musical environment.

2.3 Musical Input and Analysis

Input can be received from MIDI sources, including keyboards, electronic drums, and pitch-to-MIDI converters, analyzed in real-time, and used to direct or influence the performance by the virtual musicians. The analysis software is based on Cypher, a music analysis/generation interactive system previously developed by Robert Rowe [Rowe 1993]. IVM extends Cypher in many ways including better recognition of chords, contexts to include jazz harmony, and incorporation of real-time pattern recognition to find important melodic and rhythmic units [Rowe & Li 1995].

IVM performs a two-stage analysis of the musical input. The first stage is a lower-level analysis, yielding such properties as velocity, duration, register, harmony and chord density. The

second-stage analysis uses the low-level properties to derive more abstract, subjective attributes. These might include properties like *swingness*, *grooviness* or *drunkenness*.

As these are subjective notions, authors define these properties based on their own requirements and specify how they are computed. For instance, one might specify that a *drunkenness* property should increase as the player plays more non-harmonic notes and in less strict time. (Of course, another author might define this as *cooljazzness*.) Properties from both levels of analysis are communicated to the virtual musicians and are used to influence their performance.

2.4 Video, Sensor and Speech Input

To allow non-musicians to interact with the characters, IVM supports input from video, sensor devices and speech recognition software. Video motion detection is used to assess the movements of participants. Characters may then react to people waving their arms or dancing to the music.

IVM can integrate input from various electronic sensors. One device which we employ is an Ascension Flock of Birds, a 6-DOF magnetic position/orientation tracker from which we constructed the conductor's baton. The stroke of the baton is analyzed as the user conducts, giving control of tempo, volume and other expressive properties.

IVM incorporates speaker-independent speech recognition of North American English. This allows the characters to respond to vocal commands and spoken responses, recognizing phrases from a vocabulary of words defined for each particular scenario.

2.5 GUI Input

Graphical user interface (GUI) panels are created in Java and interface with the system by means of network communication. Panels present the user with buttons, sliders and other input tools to control any variety of performance parameters. For example, one panel may give the user control of low-level parameters such as tempo, volume and song selection. Another may present the user with sliders to tune the mood of a scene, controlling such attributes as *blueness*, *swingness*, or *hotness*. Java panels send parameter change messages (such as *tempo 120* or *swingness 0.5*) to IVM, which adjusts its parameters accordingly and communicates changes to the animation system.

2.6 Musical Generation and Output

IVM has several methods for generating, playing, processing and outputting musical performances. Score files or sequences in the form of Standard MIDI Files can be read, processed and played. Sequence processing methods include altering and transposing pitches, embellishing lines, adding inflections such as pitch bend, changing accenting and time-shifting notes.

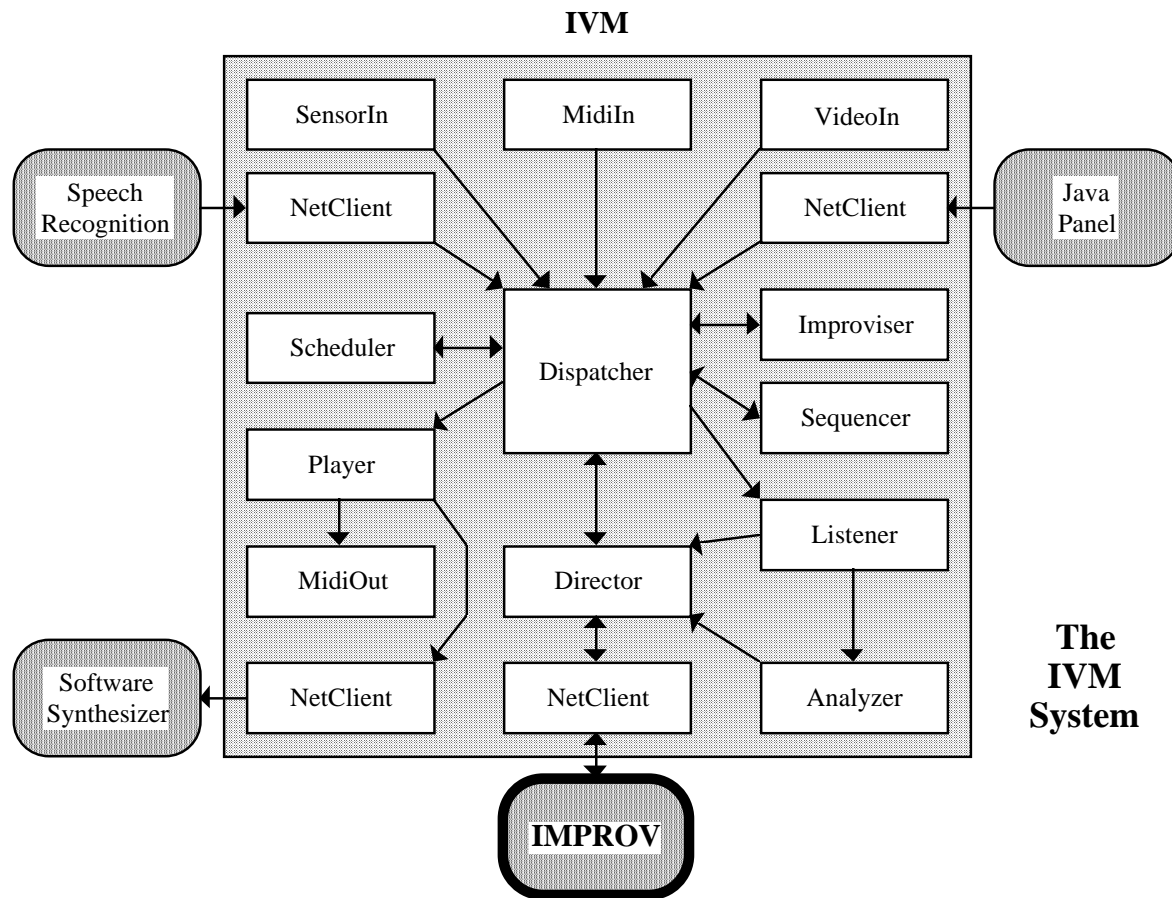


Figure 4: Diagram of the IVM system

IVM is also capable of generating improvisation. It can improvise constrained random lines and patterns based on chord changes and can also generate Markov-based improvisations. Generation methods based on Cypher are incorporated as well.

To output musical material, IVM drives both MIDI synthesizers and software synthesizers, such as Csound or NCSA's VSS. Software synthesizers are played by means of messages sent to the synthesis programs via network communication.

2.7 Animation Control

Character animation is controlled by IVM, which sends messages to IMPROV via the network. Characters are controlled on several levels simultaneously. Low-level commands control specific physical actions of the characters such as moving the fingers of a virtual saxophonist or the hands of a virtual drummer. Higher-level commands communicate information about the musical performance, user input and other environment variables. This information influences the animated performance in various ways based on each character's programmed personality.

The ability to endow characters with personalities is one of the major innovations of IMPROV. A scripting system enables authors to create decision rules which use information about an actor and its environment to determine the actor's preferences for certain actions over others. The author specifies which information is relevant and how the information is weighed in making each decision.

For example, an author might define a *nervousness* attribute for an actor which increases as other actors get closer. Furthermore, the author could specify that an increase in *nervousness* will cause the actor to choose fidgeting actions, such as shuffling its feet or biting its fingernails. Then, as other actors move closer, the character will appear to fidget more.

Using IMPROV's personality scripting, we have endowed our virtual musicians with a "body language," a set of actions which reflect various moods of playing, and given them "musical personalities" to select and control these actions. Among their many capabilities, virtual musicians can groove along to the music, tap their feet in time and raise their horns in the air during their solos.

2.8 Behavior Engine

Behind the characters' decision-making capabilities is the IMPROV Behavior Engine, a powerful artificial intelligence system. The Behavior Engine integrates numerous factors from the actor personalities, environment state and external inputs in a probabilistic decision-making process. In addition to directing actor animation, the Behavior Engine can also be used to provide feedback to the musical environment. Thus, in addition to controlling their animated performance, virtual musicians can make decisions about their musical performance as well.

Human musicians make playing decisions as individuals and as a group about how they will play a piece of music. Group decisions include tempo, volume and style of a piece. Players make individual decisions about accenting, note selection and other stylistic features. They alter their playing in response to such factors as what is played by other band members, the cues given by a conductor and the audience response.

We emulate this decision-making process in the members of the virtual band. When presented with a musical score in the form of a Standard MIDI file, virtual musicians make stylistic decisions as to how they will play the score, analogous to the way that real musicians interpret sheet music.

2.9 Implementation and Configuration

Figure 4 shows a block diagram of the complete IVM system. The IVM control software is written in C++ and runs on a Macintosh. IMPROV is implemented in Java and VRML and runs under any VRML2-compliant browser. Currently, we use CosmoPlayer on an SGI.

IVM communicates with IMPROV and other programs across a local-area network. The system makes extensive use of network communication (Telnet/TCP and UDP connections), which we have found to be a versatile and effective way to tie together system components.

The IVM control software is divided into several modules. The Dispatcher is the central control point and is responsible for receiving user input; driving the Sequencer and Improviser; scheduling musical and graphical events via the Scheduler; and routing information between other modules. The Sequencer reads Standard MIDI Files, interprets them for timing and other information and outputs MIDI events back to the Dispatcher. The Improviser generates algorithmic improvisation based on information received from the Dispatcher, likewise outputting MIDI events. The Player plays musical events received from the Dispatcher, controlling MIDI synthesizers via MIDI out ports and software synthesizers via a NetClient.

The Listener performs the first stage analysis of both live and sequenced musical input. Parameters computed by the Listener are sent to both the Analyzer and the Director. The Analyzer performs the second-stage analysis, also reporting its computed parameters to the Director. The Director communicates with the IMPROV actors. It formats messages based on information received from other modules and sends them to the actors via a NetClient. It also receives messages from the actors (generated by the Behavior Engine) and communicates this information to the Dispatcher.

We have designed the system with portability in mind and have worked to reduce machine dependence in our code. We expect that future versions will run on a single machine such as an SGI or a fast PC.

3 Conclusion

The common thread through both of these projects is a tight coupling between real-time musical analysis and the presentation of graphics and animation. Computing the ongoing behavior of musical attributes makes it possible for direct analogs between performance and image to be established both during stage performance and in unstructured installations. The challenge in realizing these projects lies in implementing the analysis and distributing the computation in such a way that the whole can be accomplished in real time. Because we use multiple computers, communications schemes become an integral part of the architecture. We have successfully used both MIDI and Telnet protocols as semaphores between music analysis/generation and graphics machines. Both of these projects demonstrate the salience of music/graphic interaction that can be achieved with real-time analysis.

4 Acknowledgments

The authors gratefully acknowledge the contributions of the following people: Doris Vila, who wrote the descriptions of the holograms and videos used in *A Flock of Words* and was responsible for the artistic conception and realization of the graphics in that work; Ken Perlin and Athomas Goldberg, for creating and developing IMPROV; Ruggero Ruscioni, for conceiving and directing the Aria project; Clilly Castiglia, for directing the IVM project; Tamara Smyth, for her help with Java; Sabrina Liao, Agnieszka Roginska and Andrea Bonotto, for their contributions to the IVM project; and the rest of our colleagues at the NYU Center for Advanced Technology, NYU Media Research Laboratory and The University of University of São Paulo, Brazil. *A Flock of Words* was partially funded by the NYU School of Education Research Challenge Fund. The Interactive Virtual Musicians project is funded by the NYU Center for Advanced Technology.



Figure 5: Gregor, an Interactive Virtual Musician

References

- [1] Canetti, E. 1984. *Masse und Macht*. English title *Crowds and Power*; translated from the German by Carol Stewart. New York: Farrar Straus Giroux.
- [2] Perlin, K. 1995 "Real Time Responsive Animation with Personality" In *IEEE Transactions on Visualization and Computer Graphics*: New York: IEEE
- [3] Perlin, K.; and Goldberg, A. 1996 "IMPROV: A System for Scripting Interactive Actors in Virtual Worlds" In *Proceedings of SIGGRAPH 96, Annual Conference Series*: New York: ACM SIGGRAPH
- [4] Reynolds, C. W. 1987 "Flocks, Herds, and Schools: A Distributed Behavioral Model", In *Computer Graphics, SIGGRAPH 87 Conference Proceedings*: New York: ACM SIGGRAPH
- [5] Rowe, R. 1993. *Interactive Music Systems: Machine Listening and Composing*. Cambridge, Mass.: The MIT Press.
- [6] Rowe, R.; and Li, T.C. 1995. "Pattern Processing in Music" In *Proceedings, Fifth Biennial Symposium for Arts and Technology*: Center for Arts and Technology at Connecticut College
- [7] Rowe, R.; Singer, E.L.; and Vila, D. 1996. "A Flock of Words: Real-Time Animation and Video Controlled by Algorithmic Music Analysis" In *Visual Proceedings, SIGGRAPH 96*: New York: ACM SIGGRAPH.
- [8] Singer, E.L.; Perlin, K.; and Castiglia, C. 1996. "Real-Time Responsive Synthetic Dancers and Musicians" In *Visual Proceedings, SIGGRAPH 96*: New York: ACM SIGGRAPH.
- [9] Singer, E.L.; Goldberg, A.; Perlin, K.; Castiglia, C.; and Liao, S. 1997 "Improv: Interactive Improvisational Animation and Music" ISEA 96 Proceedings, Seventh International Symposium on Electronic Art: Rotterdam, Netherlands: ISEA96 Foundation