Generation and Implementation of Mixed-Reality, Narrative Performances Involving Robotic Actors

Nikitas M. Sgouros and Sophia Kousidou

Department of Informatics, University of Piraeus, Karaoli & Dimitriou 80, 18534 Piraeus, Greece
{sgouros, nova}@unipi.gr

Abstract. Recent advances in robotics and multimedia technologies have created new possibilities for staging narrative performances involving robotic actors. However, the implementation of these types of events is complicated by the lack of appropriate direction and execution environments that can deal with the complexity of these productions. This paper seeks to address this problem by describing CHOROS, a Java-based environment for authoring, direction and control of narrative performances. In particular, CHOROS allows the story author to annotate the performance script with stage directions. Furthermore, the environment offers to the performance director an augmented reality interface for planning the behavior of the actors. Finally, the system uses vision-based tracking methods and behavior-based control for adjusting the behavior of the robotic actors according to the director instructions during the performance.

1 Introduction

Recent advances in robotics and multimedia technologies have created new possibilities for integrating robotic actors in narrative performances. Such a development greatly enhances the means of expression available to creators of storytelling environments by allowing them to stage mixed reality events in which human and robotic actors along with various other multimedia objects strive to create immersive and enjoyable narrative experiences. Unfortunately, the development of these types of performances is hampered by the lack of appropriate directing and execution environments that can deal with the complexity of conceiving and the unpredictability of executing these projects. The majority of these productions uses a mixture of traditional multimedia authoring tools and robot programming environments that can deal only with isolated aspects of the development and execution process. In particular, traditional multimedia authoring and presentation tools can only describe the spatial and temporal relationships between the multimedia objects (i.e., video, audio, graphics) that comprise a multimedia application. These tools lack the ability to automatically track the behavior of human or robotic actors and associate it with the execution of various multimedia objects during the staging of a narrative event. On the other hand, current robot programming environments are not able to describe and execute

O. Balet, G. Subsol, and P. Torguet (Eds.): ICVS 2001, LNCS 2197, pp. 69–78, 2001.

behaviors that should be synchronized with the rendering of various multimedia objects (e.g., synthetic actors, speech, video or audio clips). In addition, the appearance and behavior of most current mobile robots is not expressive enough to support acting. Consequently, a new generation of development and execution environments for narrative performances featuring robotic actors needs to be developed so that:

- story authors can incorporate stage directions in the narrative text
- directors can plan off-line the behavior of all the actors in a performance
- actor behavior can be automatically tracked and guided during the staging of the performance according to the director plan
- the behavior of the robotic actors can be expressive enough in order to support acting.

This paper describes CHOROS, a development and execution environment for narrative performances featuring human and robotic actors that seeks to address these requirements. At its current stage of development the system provides assistance to the author and director of a narrative performance. In addition, it provides a run-time environment that monitors the implementation of the directing plan during the performance.

In particular, CHOROS allows the story author to incorporate stage directions in the performance script by annotating the spoken dialogue between the actors with prosodic information using the Java Speech Markup Language (JSML). In the case of the robotic actors this information is fed to a speech synthesizer that verbalizes appropriately the annotated text during the performance.

The environment offers to the story director an augmented reality interface for drawing the paths that will be followed by the performance actors. Planning is performed on a top-level view of the actual performance space as captured by an overhead video camera. We refer to this space as the *stage* for the event. The director can specify graphically the temporal constraints describing the use of synthetic, audio or video objects at specific regions in these paths by ordering these objects in *path-bound timelines*. These are timelines that are drawn in parallel with the actor paths on the stage view. We refer to the specified paths and timelines as the *director plan* for the narrative performance. The environment automatically analyzes the director plan to extract a set of both qualitative and numerical constraints on the spatial and temporal behavior of the actors during the execution of the application. The numerical constraints include the motion parameters for the robots (speed, acceleration etc.) that are theoretically necessary in order to follow their designated trajectories and synchronize their movement with the rest of the multimedia objects. The qualitative constraints include the groups of actor motions that should be executed concurrently during the performance based on their media constraints along with the spatial relations between them (e.g. parallel, converging, diverging etc.). CHOROS provides the director with a simulation environment in which s/he can visualize and monitor the execution of his/her plan in the event stage.

At run-time, all the original and extracted constraints are fed to an execution module that constantly tracks and adjusts the behavior of the robots in order to follow the director plan. Tracking is using frame-differencing operations on a grid-based decomposition of the stage view to detect the position of each actor in space and analyze the direction and speed of its motion. Adjustment seeks to deal with the unpredictability of controlling robot movement at run-time. This goal is achieved by mainly preserving the qualitative constraints between the behavior of the actors and the rest of the multimedia objects in the face of frequent deviations of the actors from their designated trajectories. These deviations are caused by either sensor or actuator errors on the robots or the approximate interpretation of the scenario by the human actors.

CHOROS can be used for the development of narrative or more general multimedia performances that involve robots such as robotic theatre productions or puppetry, interactive playspaces [1], dance productions, programming of tour-guiding or entertainment robots and creation of robotics-based special effects for the movies.

2 The Directing Process

Directing environments for storytelling environments involving robotic actors need to provide intuitive methods for describing the behavior of these actors. For this reason, the directing process in CHOROS uses a mixture of augmented reality and timeline-based techniques.

In particular, the system provides the director with an augmented reality environment in which the trajectory of each robot can be drawn on a top-level view of the stage. In this case, the path planned for each robot consists of a sequence of line segments. The director determines the starting and ending points for each segment by clicking on the desired points on the screen. These points can signify either a change in the direction of movement of an actor or the enactment of constraints associating the spatial location of the actor with the rendering state of various multimedia objects. In the second case, the director specifies the frame number of a video object or the time position of an audio clip that should be rendered whenever the robot reaches the particular point in its trajectory. In addition, the director can specify the piece of text that should be verbalized by the speech synthesizer whenever the robot reaches such a point. Since frequent sensor or actuator errors by the robots make it very difficult to achieve an exact synchronization of this sort at run-time, the director is able to specify a region in space centered on the specific point in which the particular constraint should be satisfied. Once such a region has been specified for both ends of a line segment, the system draws a path-bound timeline parallel to this segment. This timeline depicts the starting and ending positions along with a set of intermediate positions of the multimedia object that will be rendered while the actor follows the specific line segment. The existence of such a timeline provides to the director an effective way of visualizing the association between the behavior of the robots and the rendering of various multimedia objects. In addition, this timeline allows the developer to monitor effectively the synchronization between the actors and the multimedia objects at execution time.

For example, Figure 1 provides a snapshot of the stage view in CHOROS that contains two robotic actors. The stage floor in CHOROS is covered with a black material in order to facilitate real-time tracking of the behavior of the actors. The figure depicts the path planned for each actor and its associated timeline, which, in this case, refers to a video object for each actor. The rectangular areas in the path of each actor describe the association of a particular region in the trajectory of the actor with a specific rendering state of a multimedia object. In the figure, we refer to such an association as a media-driven constraint.

The resulting directing environment seeks to describe the major ways of associating actor behavior with the use of various multimedia objects in a performance. More specifically, such an association can be either *actor-driven* or *media-driven*. In the first case the behavior of the robots in the stage drives the use of the multimedia objects.

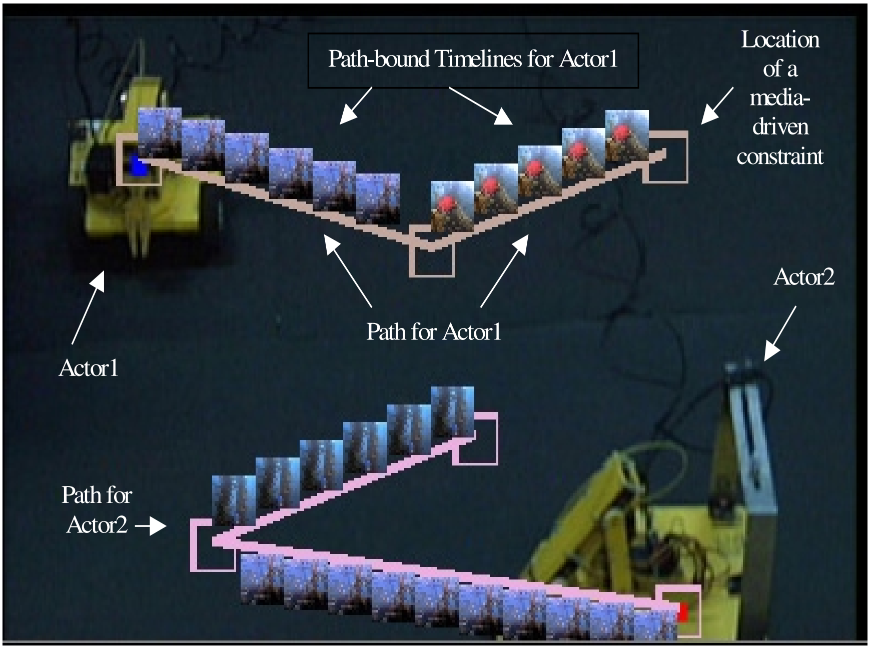

Fig. 1. The stage view in CHOROS.

For example, an actor in a robotic theater production can verbalize a particular text segment whenever it approaches certain areas in the stage. On the other hand, a media-driven association is one in which the execution of the multimedia objects dictates the behavior of the robotic actors. For example, the music in a dance performance usually forms the basis for organizing the movement of the robots participating in it.

2.1 Specification of Actor-Driven Associations

The directing process supports two types of actor-driven associations. The first one covers *location-specific* cases in which the execution of a set of multimedia objects starts or terminates when one or more robotic actors enter or exit from specific areas in the performance space. These areas can be either static, as in the case of the stage area in the robotic theater example, or mobile, such as the region surrounding another human or robotic actor during a performance. In the second case, speech, video or audio clips are triggered during the negotiation of the space that separates the actors. For example, a robot (e.g. R) might execute appropriate audio clips (e.g., sound a horn) that warn another actor (e.g. A) against approaching it. These clips are activated whenever A enters a region designated by the author that constantly surrounds R.

The director is able to describe location-specific interactions by indicating in the augmented reality environment specific areas in the performance space and associating the execution of multimedia objects with certain actors entering or exiting these areas. At the time of definition, the system checks whether these areas contain another actor. If so, the area becomes bound with the robot and follows it through space, otherwise the area is considered to be static.

The second type of association covers *behavior-specific* cases during which a particular sequence of commands executed by a robotic actor should trigger or terminate the execution of a set of multimedia objects. For example, when an artificial pet wags his tail a series of audio clips from real pets engaging in this behavior can be rendered in order to reinforce the illusion of a real pet. We are currently developing a graphical environment for associating sequences of motion commands of an actor to the enactment of constraints affecting the execution of multimedia objects. This environment will allow the director to associate reflex-like behaviors of the robots with appropriate multimedia objects.

2.2 Specification of Media-Driven Associations

The environment identifies automatically media-driven associations in the behavior of a particular actor (e.g. R) through the detection of *media-driven segments* in its path. A media-driven segment for R is a sequence of line segments that contain in its starting and final points a constraint on the use of the same multimedia object. For example if the author has specified that at point A in the path for actor R video object V should be in frame position F_1 and at point B in the same path V should be in frame position F_2 then the sequence of line segments in this path that start from A and end in B form a media-driven segment in the trajectory of R.

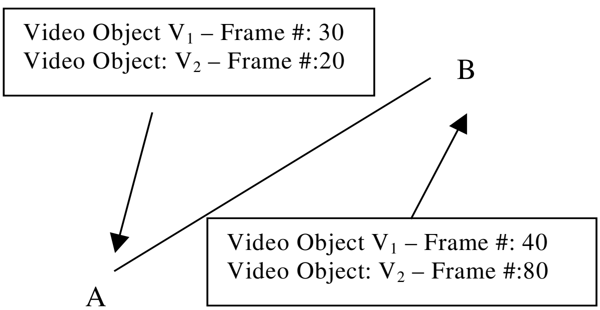

Fig. 2. Example of an inconsistent media-driven segment.

Once the system has detected a media-driven segment it checks whether it is *consistent*. A consistent media-driven segment for robot R is one for which there is a set of motion parameters (e.g. speed) for R that allows it to follow its designated trajectory in the segment and satisfy all the constraints at its starting and finishing points. For example, if we assume that video objects V_1 and V_2 should be rendered with the same frame rate then Figure 2 depicts an inconsistent media-driven segment because there is no speed for the actor that will allow it to traverse the segment and satisfy the constraints for both video objects V1 and V2. The system notifies the author of inconsistent line segments in order to take remedial action.

2.3 Qualitative Analysis of Media-Driven Segments

The analysis of media-driven segments seeks to extract a set of qualitative constraints that describe the spatial and temporal relations between the behavior of the actors during the performance. Analysis proceeds through the execution of the following sequence of steps:

1. Detection of concurrent points and segments in the actor paths.
2. Extraction of qualitative spatial constraints on concurrent segments.

Detection of concurrent segments. Two points in the paths for two actors (e.g. R_1 and R_2) are *concurrent* if they are extreme points (i.e., starting or final points) in their respective media-driven segments and they share at least one constraint.

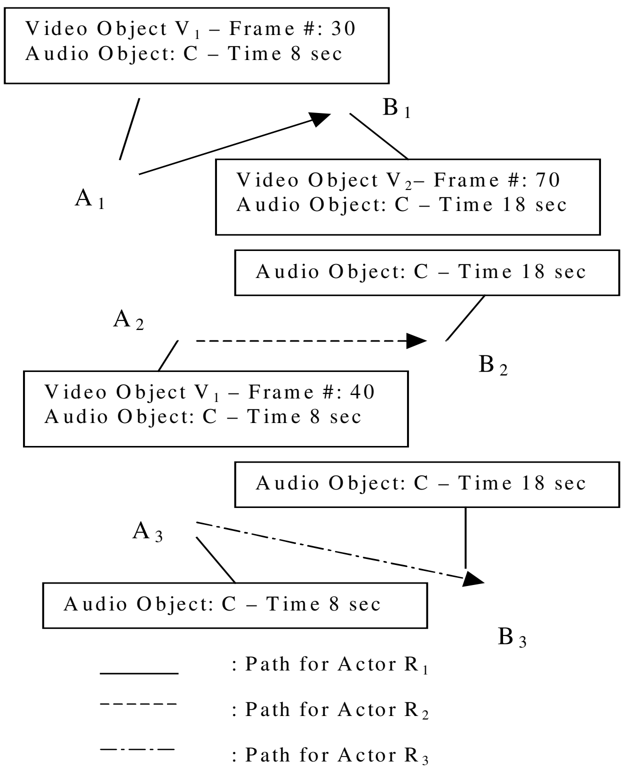

Fig. 3. Examples of concurrent points and segments in media-driven associations.

For example, in Figure 3 the points A_1 and A_2 in the paths for actors R_1 and R_2 are concurrent since they share the same constraint for audio object C.

Two media-driven segments for two actors (e.g. R_1 and R_2) are *concurrent* if they have concurrent extreme points. For example, in Figure 3 the segments A_1B_1, A_2B_2 and A_3B_3 that correspond to actors R_1, R_2 and R_3, respectively, are concurrent because their extreme points A_1, A_2 and A_3 along with B_1, B_2 and B_3 are pair-wise concurrent.

For each multimedia object the authoring environment lists the media-driven segments that are constrained by it. For each such object the set of media-driven segments that are pair-wise concurrent form a *concurrent set*. For example, in Figure 3 a concurrent set for audio object A is: { A_1B_1, A_2B_2, A_3B_3 }.

Extraction of qualitative spatial constraints. During media-driven interaction it is often the case that the lines and shapes each actor produces through space can be related to those of other actors through copying, complementing or contrasting relations. The establishment of these relations creates a momentary image which holds meaning for the audience. The purpose of this step is to extract a set of qualitative spatial constraints that describe the orchestration of the movement of all actors. Currently, the line segments in each concurrent set are classified as:

1. Parallel, if they have approximately the same slopes. Two parallel segments can be *opposite* if the actors involved move in opposite directions or *analogous* if they move in the same direction.
2. Converging, if the distance between their end points is less than a user-specified threshold while their starting points are further apart.
3. Diverging, if the distance between their starting points is less than a user-specified threshold while their end points are further apart.

3 The Execution Process

The execution process constantly tracks and adjusts the behavior of the robotic actors in order to follow the director plan. Tracking is using background separation and frame-differencing operations on a grid-based decomposition of the stage view to detect the position of each actor in space and determine the direction of its motion and its speed. Adjustment seeks to deal with the unpredictability of controlling robot motion at run-time. This goal is achieved by mainly preserving the qualitative constraints governing the behavior of the robots.

3.1 Actor Tracking & Guidance

The tracking process accepts as input the stage view and assumes that the background of this view will remain constant at run-time. This allows it to perform background separation at each frame and then compute the difference between successive frames. Each one of the resulting frames is then mapped to a grid of twelve cells. Each grid cell is assigned a number, which is equal to the number of interesting pixels in it, i.e., the pixels with values above a noise threshold. This threshold has been computed during a calibration procedure for the particular stage view. Furthermore, for each grid cell with a positive number, the method computes the center of gravity of its interesting pixels. In order to determine the location of each actor in the stage view the tracking process picks the grid cells with the highest numbers and uses a minimum distance classifier which assigns each center of gravity in these cells to a particular class that represents an individual robot. This particular tracking process has achieved a success rate of over 90% in CHOROS.

The results of the tracking process are fed to a guidance module that forces each robot to follow as closely as possible the current line segment in its path. In particular, this module constantly issues a set of motion commands to each robot that seek to minimize the distance of the current location of the actor from the final point of its current segment in the stage view. At each point in its trajectory these commands move the robot in a direction approximating the direction of the line connecting its current location in the stage view to the final point of the current line segment in the same view.

3.2 Behavior Adjustment

Behavior adjustment accepts as input the numerical and qualitative constraints describing the association between the behavior of the robots and the use of various multimedia objects. It seeks to preserve these associations by issuing appropriate motion commands to the robotic actors.

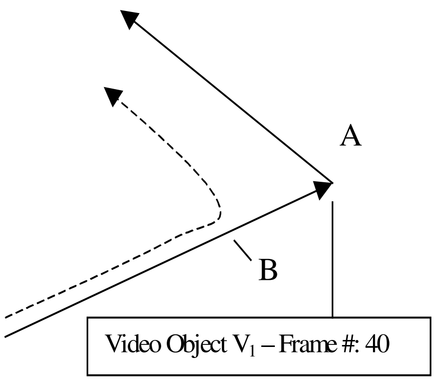

Fig. 4. Change the behavior of an actor as soon as the relevant constraint on a multimedia object is satisfied.

Because frequent sensor or actuator errors make it almost impossible to satisfy the numerical constraints on all segments of an actor path, the adjustment process seeks to preserve primarily the qualitative constraints associating the traversal of each segment with the rendering of various multimedia objects. To this end, the process applies the following rules:

Whenever a multimedia object reaches a rendering state that has been associated with a change in the behavior of an actor in a media-driven segment, then this change will take place irrespective of the location of the actor in the stage.

For example, Figure 4 depicts the desired and the actual trajectories that will be followed by an actor, assuming that video object V_1 reached frame 40 at point B and not at point A as it was specified during authoring. The desired trajectory is drawn with a continuous line, while the actual trajectory is drawn with a dotted one. In this case the actor will actually turn left at point B.

If an actor reaches a location in space before a multimedia object reaches a rendering state that has been associated with this location then the actor will remain in this location until the desired rendering state is reached.

In Figure 4, for example, if the actor reached point A before video object V_1 reached its 40^{th} frame then the actor will remain in A until frame 40 is rendered. It will then continue to follow its specified path.

Both rules ensure that concurrent media-driven segments will produce concurrent behaviors at run-time. Consequently, the temporal structure of the behavior of the actors that was prescribed during the authoring phase will be preserved. However, the application of the first rule will not preserve the exact spatial structure of the behavior of the actors. In order to preserve the qualitative constraints on the actor movements, the execution process applies the following rule:

If a group of robotic actors begins to follow a concurrent set of media-driven segments then the system will try to satisfy the qualitative spatial relations, if any, between the elements of this set. In particular, parallel segments must remain parallel, while convergence or divergence relations should be preserved between the elements of this set.

Consequently, if there are deviations between the actual starting position of a segment in the concurrent set and its prescribed starting position from the authoring phase, the system will determine a new final position for this segment. The computation of these new segments will take place in the stage view and it will try to satisfy the geometric relations between the elements of the set. The length of the new segments should not exceed the length of their respective segments from the authoring phase in order to ensure that the media constraints at their end points can be reached.

4 Implementation

CHOROS has been coded in Java using the Java Media Framework API for dealing with the multimedia objects and the Java Communications API for managing the robots. The robots used in CHOROS consist of a pair of low-cost mobile manipulator kits from Lynxmotion that communicate with a Pentium II PC using its two serial ports. The robots carry no sensors. The only sensor that is used by the system is an overhead video camera connected to the PC that provides a 320 x 240 stage view for the application.

Up to this date, CHOROS has been used in a series of trials implementing media-driven associations between the behavior of the robots and the rendering of speech, audio or video objects.

5 Conclusions, Related and Future Work

This paper describes a direction and execution environment for narrative performances featuring autonomous mobile robots. This work extends research on interactive playspaces [1] by allowing their integration with robotic actors and providing appropriate development environments for them. In addition, it seeks to support the creation of a new generation of authoring and execution systems for interactive narrative performances that coordinate the interaction between physical actors (i.e., robots or humans) based on higher level plot structures and/or audience reactions [2-3].

Future work in this area will focus on extending the means of expression of the director during the authoring phase. This will be achieved through the development of a rich vocabulary for composing movement and linking it with the rendering of various multimedia objects. Systems for analyzing and transcribing movement, such as the Laban or Benesh notations [4-5], can provide inspiration for implementing these types of extensions. Furthermore, future research will seek to implement expressive behaviors in the robotic actors that are suitable for acting.

In terms of content development, CHOROS is currently being used for the creation of a robotic theater production in conjunction with Yiannis Melanitis [6], an artist working on robotic performances. In this production, the system is used for planning and controlling the behavior of two robotic actors, a hexapod and a mobile manipuilator. In addition, CHOROS controls the movement of a pair of robotic cameras that move in the performance stage in order to capture the development of the event according to the director instructions and broadcast it on the Web.

Acknowledgements. This work was supported by the Greek Secretariat for Research and Technology under the program «Hygeiorobot» (GSRT-PENED 99).

References

1. Bobick, A. F. et al, The KidsRoom, *Communications of the ACM*, vol. 43, no. 3, March 2000.
2. Sgouros, N. M., Dynamic generation, management and resolution of interactive plots, *Artificial Intelligence*, 12(3), 29-62, Elsevier (1999).
3. Sgouros, N. M., Detection, Analysis and Rendering of Audience Reactions in Distributed Multimedia Performances,» *ACM Multimedia 2000*, Los Angeles, CA, November 2000.
4. Laban, R., *Principles of Dance and Movement Notation*, New York: Dance Horizons London: MacDonald & Evans, 1996.
5. Blom, L. A., Chaplin, L. T., *The Intimate Act of Choreography*, Dance Books Ltd., London UK, 1989.
6. http://digitalart.asfa.gr/students/melanitis