Consiglio Nazionale delle Ricerche

CATS
Computer Aided Theatrical Score
Esprit Project 20340

A study for the user staging model of CATS

Patrizia Palamidese

CNUCE C97-022

CNUCE

CNUCE Internal Report, C97-022

A study for the user staging model of CATS

Patrizia Palamidese

Abstract

This report describes an animation system to design actors' actions, compose them with the other elements of a multimedia scene, and obtain a system of synchronized events. We have analysed current staging methodologies used in various fields to find a set of common concepts for defining a general staging model, valid for different kinds of users and different applications. Obtaining a good degree of generality and completeness entailed an iterative work of specification, prototyping, and testing in collaboration with professionals in the artistic field. Unlike traditional systems which treat media in separate software, for example human body motion only or speech only, we consider the scene elements all together according to the theatrical paradigm which is the basic framework for describing a performance. Our model applies to each kind of spatial-temporal representation including television, cinema scenes, or events that take place, for example, in town squares. It addresses the needs of multimedia scenes builders who are not computer experts. This includes particularly directors, artists, managers or technicians in the fields of theatre, cinema, or TV, who have to plan and evaluate the spatial and temporal arrangements of actors with other media in a virtual scenario before their practical realization on a real stage.

Keywords: human figures animation, multimedia scenes, multimedia script, staging

1. Introduction

In this report we describe a computer aided staging model to design characters playing in multimedia scenes. We provide a homogeneous and integrated approach for specifying not only characters' actions but also for defining the other elements which usually belong to a scene such as characters' speech, music, sounds, set, lights and camera motions. Basically, users can make a three-dimensional sketch of their staging idea, define, simulate and modify single elements, assemble and synchronize all of them into a unique composition. They can also run a 3D multimedia simulation of their *mise en scène* to evaluate the artistic result and the characteristics of its actualization.

To build a complete and general *user model* we analyzed the practical work of theatrical directors while they were preparing a performance but we found it very difficult to make a synthesis due to the lack of standard practices in this field. Since theatre is an art, the creation of a *mise en scène* has always been considered the result of artistic creativity and improvisation rather than an industrial product. Therefore there is no system of stable rules to describe the staging process even though staging is an activity which has been practised for a long time.

Only in more recent times, has one been able to assist in efforts for planning and formalizing the direction of a performance. This includes making written plans and/or relying, for some activities, on existing general purpose software such as computer aided design systems, image processing tools, or, more simply, office automation tools. Only a few specialistic tools exist which cover lighting design.

The main work for preparing a production consists in creating the script which describes everything that has to be done and contains comments and drawings which explain everything in detail. Though the script is universally used, it cannot be assumed as a model for implementation because it neither has a standard format nor does it provide unequivocal elements for defining a specification language. In practice the original text, which contains the dialogues between the characters, is progressively refined and enriched with an exhaustive description for each element which has to be built or executed: it can be, for example, a piece of furniture for the set, the gesture of an actor, the tone of the voice, or the change of a light. Each member of the cast, whether s/he is an actor or a technician, personally takes notes of the things s/he has to do and adds explanations and instructions to the text in the form of hand written annotations and drawings.

In principle each theatrical element is a *system of signs* with its specific grammar, but, in practice, it is not governed by written rules, rather it is described by concepts which are transmitted from artist to artist verbally. Only *music* has a standard notation system which is universally understood so that any musician can interpret and represent the musical score even if it has been written by someone else. As for the other elements of a scene, a standard specification method which is universally accepted doesn't exist. The field where most attempts have been made to introduce a formalism and a standardization are human body motion and choreography; also camera, lights and sounds are sometimes described by sets of symbols which seem to have been accepted by several groups, though it is not possible to really talk of standard notation (Palamidese, Tofano & Scapellato, 1996).

Labanotation (Hutchinson, 1970) Eshkol-Wachmann (Eshkol & Wachmann, 1958) (Feldenkrais & Eshkol,), and Benesh notation (Benesh, 1956) are well known symbolic methods invented by choreographers for describing the position and orientation of body parts. Since the use of these movement notation systems is inherently complex a number of computer tools were developed for editing and interpreting them, mainly in the 1970s (Smoliar & Weber, 1977) (Sealey, 1979) (Wolofsky, 1974) (Barenholtz, Wolofsky, Ganapathy, Calvert & O'Hara, 1977) (Calvert & Chapman, 1978). The result of these experiments is summarized by Calvert (Calvert, Bruderlin, Mah, Schiphorst & Welman, 1993) who has acquired considerable experience in working with dancers and choreographers. He claims that these systems have an important role as archival tools in recording dance, but are not very useful as tools for supporting the compositional process. Therefore he follows a different strategy to develop a completely new system, LifeForms (Calvert, Bruderlin, Dill, Schiphorst & Welman, 1993), which provides a creative design support for the conception and development of dance. A deep understanding of the mental model of the choreographer's design process was necessary to build the LifeForms computer interface and make it compatible with a choreographer's design skill. LifeForms took six years to develop (Calvert et al., 1993) including collaboration with choreographers, during which the system has been revised and improved several times. User feedback has affected not only how the functionality is presented to the user but, sometimes a new concept has required a reimplementation of the program.

The tasks a choreographer has to perform are not exactly the same tasks as those of a director or a scene composer. In principle a choreographer starts from the movements of a dancer and goon adjusting it until his postures are correct with respect to the other dancers in each moment of the ballet, while a director starts from an event and plans all the events that follow and precede it. The main difference is that choreographers have to keep a detailed control on human body movements while directors are deeply concerned with the composition of macro elements.

Notice that the ideal animation software will include a complete set of capabilities to treat the problem at many levels, from a very abstract to a very accurate point of view. Such software contains a hierarchy of levels which allow users to make a preliminary outline of the performance and then to detail each element more and more accurately down to the level, for example, of human body joints. Also, the composition of events will be meticulous in order of getting an effective and realistic result: for example, within a speech one can specify that a particular voice inflexion has to correspond to a specific gesture. However, a fine grain specification and synchronization though enhances the level of realism, is obtained to the detriment of the user who has to dedicate a lot of time (weeks, perhaps months) to get it.

The system described in this report supports a new design model for the conception and development of scenes and whole performances at a high level of abstraction. It allows to: a) support the conception of a performance, b) make a quick draft of main ideas, c) evaluate the result in real time, d) manipulate symbolic descriptions of scenes and performances, e) synchronize events.

The staging model we describe has been first partially implemented into the CATS Maquette, then into the Character Animation tool of CATS (CharTool) and, finally, has been transferred to CATS. In the course of this report the description of the user interface model is supported by references to the CharTool implementation. CharTool is a research testbed for actor movements implemented in Visual C++ and OpenInventor. It includes a C++ library of basic classes to manage actors' movements, and a higher level module which contains the semantics of the user interaction. This library is transferable without changes to CATS. CharTool has an autonomous user interface developed in Visual Basic. This tool presents a basic set of functionality which can be easily expanded with new higher-level and lower-level animation capabilities up to create a complete animation system for human figures, cameras, and lights.

The user interaction model draws its inspiration from the theatrical paradigm and also from virtual reality technology. A virtual reality system is a system which makes the interaction between human beings and objects as similar as possible to such interaction in the real world (Earnshaw, Gigante & Jones, 1993) (Thalmann, 1993). Our user model is centred around the ideas of the stage and actions that take place on it. Though the current implementation doesn't exploit virtual reality devices such as, for example, stereo view and data gloves, the concepts described in this report are applicable to a more advanced architecture.

In this report we first describe the performance design process which has inspired our system, then we illustrate the user staging model that we have defined to support this activity and its main components: the *multimedia script* and the*virtual rehearsal*.

2. The performance design process

Initially we tried to identify the tasks that directors accomplish when they plan a performance. Unfortunately staging is not a field which offers a software engineer an organized set of standard concepts since the production of a play is artistic and creative work which is not governed by fixed rules that are common to all. Artists rightly reject any attempt to classify and categorize their work and directors defend the originality of their approaches and methods. They rely more on their creativity during rehearsals and on decisions taken on the spur of the moment rather than on accurately planning everything in advance.

Thus, in the first phase of this project we were only able to perform a high level analysis of the production process with the help of directors and other professionals working in theatre, cinema, and television. Only later were we able to understand and deduce from this data the essential characteristics of the performance design process. We summarize here the conclusive data of our work.

In theatre, cinema, and TV, the steps of the production process are slightly different and, in addition, the activities are described with different terms and concepts even when they are similar. All the information collected during the preliminary discussions is reported in (Palamidese et al., 1996). After an accurate analysis we managed to condense this material into a homogeneous though schematic description which divides the production process into three phases: *pre-production, production*, and *post-production*. It was not easy to identify the activities contained in these phases. Only many discussions helped us to find out the macro tasks they include.

Pre-production starts after the choice of the script, or the show format in TV, and is the mandatory moment for making a design of the performance. People start by analysing and elaborating the text before making executive plans for each component of the performance. It is a cyclic process which is repeated until the director or the theatre manager gives the final approval: first people elaborate a proposal (for example designs for sets or lights), and then submit it to artistic and cost evaluation. The main result of all this work is a script which illustrates what has to be done. In cinema this work is completed by the storyboard which is a sort of preview of camera shooting.

Though the production phase is really different from field to field and does not take the same time in theatre, cinema and TV, nevertheless we can say that it always includes rehearsals and script revision. During rehearsals the original script is subject to modifications and someone in the team is in charge of it (for example the *script girl*, in cinema). Another common necessity is to play all the elements altogether to check and tune their synchronization. In reality this control, though fundamental, is often left to the end due to lack of time. It should be noted that even the change of a scene, in theatre, which includes assembling and disassembling the set, needs to be rehearsed and synchronized to check its duration. Note that production, in the movie industry, is the phase when the real shooting is made; instead in TV it includes broadcasting.

Though post-production should really deal with the distribution of the finished product, there are still some reasons to rearrange the script. During the *home exhibition* of a theatrical performance, i.e. when it is performed in the theatre for which it was designed, the play might undergo some changes which have to be annotated and memorized in the script. Again a theatrical play is often run in other theatres (*turnée*) or environments which might have a different size and/or shape from the original place. The play thus has to be adapted to the new situation by changing, for example, the set, or the movements of actors on the stage. As for TV, at the end of each broadcast the basic scheme of a program might be corrected to improve the quality of the result.

3. General features

Directors and artists are involved in the *staging conception* on several occasions during the production, however the activities they perform can be reduced to two major ones: the elaboration of the *script* which describes the structure of the show and all its components, and the creation of the *live show* on stage (*rehearsals*). Rehearsals are a means for evaluating and reviewing the script as they map to real time and space what was previously conceived and written. Under these aspects directors' tasks are similar, whether they are theatrical, television, or cinema directors; and there is no reason to design different tools. Other considerations support this thesis.

With the collaboration of professionals from all fields we have examined several kinds of performances, from plays to operas, from TV dramas and shows, to parades. Whichever scene is built, it always contains the same kinds of elements and the composition techniques are alike. For example, they need rules to make contemporary events (*while A then B*) or sequential events (*A starts when B finishes*). These techniques are only used with different styles in theatre, cinema, and TV. Each element, in fact, is governed by its specific theatrical, cinema, or television language, which usually varies not only from field to field but also from director to director.

Only the concepts of camera and point of view may seem discriminating at a first glance. Theatrical directors need to check the scene from the audience's point of view. Hence they might want either to move the point of view interactively (in a static scene or during the simulation) or position a number of *eyes* in strategic positions around the theatre. In cinema, as a single camera is used which follows the actors moving in the set, a scene has to be repeated many times to take it from different angles. For example a dialogue between two people is taken twice: in one shot the camera focuses on the first actor, in the other shot it focuses on the second actor. Only in the editing phase will the director build the right sequence of shots. In a computer system one can follow the same approach by defining two cameras. Instead all the live performances such as television shows are taken simultaneously by many cameras. Thus, a computer system which includes a multiple camera control and allows one camera or another be activated during the rendering process, will cover all such needs.

With the existing animation systems only unstructured sequences of scenes can be built and there's no support for specifying a whole performance as described in the script. This is a big limitation because the architecture of the performance is the first level of concretization when users pass from the idea to the design. Our system maintains this logical structure which can be continuously revised by users who can thus experiment with different realizations of their ideas.

Our system doesn't aim to make a realistic production but to build a staging model which provides instructions either for realizing a real performance or a virtual one (in a virtual studio (Breiteneder et al., 1996) (Gibbs, 1996)) or a mixed real-virtual (for example real actors in a virtual set). Therefore we have two main constraints: little time for designing a scene, and simulation in real time. To drastically reduce the time necessary to create actors' animations the system provides a high level specification interface with which users compose pre-defined movement patterns; thus they are able to plan the key actions of actors, since a realistic representation will require much more specification work. On the other hand simplified 3D objects and actors provides a symbolic representation which can be rendered in real time on a low-medium level machine. Geometric and animation characteristics will suggest the temporal and spatial presence of 3D objects and actors, rather than show detailed aspects.

4. The multimedia script

Unlike animation systems where users aim to create a well defined product, CATS users aim to plan the construction of their product which will be built later on either in the real world or in a more sophisticated virtual world. Hence CATS is a computer aided design system. Users aim to create the *multimedia script* of the performance which is the computer equivalent of the traditional script but where hand written annotations are replaced by executable properties of objects. The multimedia script contains both the logical structure of the performance and the detailed descriptions of the individual components. Physically it resides in a hierarchical data structure which is maintained by the system and is interactively modified by the user through the user-computer interface. Users can change both the layout of the performance and individual components.

We provide different ways to represent the multimedia script so that users can analyse and modify it under different perspectives. The most traditional way to deal with it is the *layout*, which shows the interdependency of the components between them. Alternatively users can run a *simulation* and play a series of actions in real time. Finally, they can work with the *score* where all the dynamic elements of a scene are represented by symbols.

4.1 Layout

We have defined a unique layout for theatrical and cinema applications. The following table shows to what extent the components of a performance are similar and different if theatrical and cinema styles are followed. Notice that television style is included in the theatrical category. The evaluation factors are spatial, temporal and narrative continuity. The main difference is that in theatre, scenes and actions are characterized by space continuity, while in cinema a scene (and even a shot) may occur in different places. For performance and memory reasons we have decided to adhere to the theatrical philosophy and assumes that a scene takes place in a circumscribed space so that a geometrically well defined unique set is used. This choice fits the requirements of the shooting phase in cinema as well since it takes place in a set (given a set they make all the shots related to that set), but it doesn't completely describe a movie in the editing phase when a scene may consist of shots taken in different places. With this limitation, we can assume that our performance structure is general enough to be applicable to most applications.

The *layout* (Figure layout) describes the performance in terms of acts (parts), scenes, and actions (shots) and is represented on the screen by a tree. As many branches as there are acts depart from the root (performance). An act can be expanded into scenes. A scene contains two types of information: the list of characters and the list of actions. Finally, actions can be expanded into the list of items which determine it: for example the *action1* might require lights *(l1, l2)* and the music *music1*. By selecting an item a notebook appears where the user might want to write either personal comments such as *change from day light to night light* or detailed information on the item such as *this light is a spotlight.*

Comparative table for theatre and cinema performances.
s, t, and n indicates spatial, temporal and narrative continuity respectively.

Theatre/CATS		Cinema	
Line	s, t, n	Line	s, t, n
Action	s, t, n	Shot	(s), t, n
Scene	s, t, n	Scene	(s), t, n
Block	n	Sequence	n
Act	n	Part	n
Script	n	Script	n

These components of a performance can be manipulated by users:

Line or Text or Cue. This is the smallest unit of the composition and corresponds to a phrase spoken by the actor (*cue*) in a dialogue. Our system memorizes both the text and the voice that pronounces it, along with timing information.

Action. In a theatrical representation the scene is divided into actions each with a specific dynamic and stylistic connotation - a dialogue, a piece of music.

Shot. In cinema and TV the basic unit is the shot which is the phase when a camera is filming uninterruptedly from the beginning to the end. During a shot the camera doesn't necessarily move but can make a fixed take. Let us consider this example: the camera starts from the door and moves to an actor sitting on a chair, then it stops and makes a close up for a while. There are two shots here: in the first shot the camera is moving, in the latter the camera is fixed. A single shot can be very long. For example the operator uses a steady camera to shoot all the rooms of a house where a famous person lived. S/he makes a subjective take which moves from object to object, from room to room to the garden. This is a unique, uninterrupted, long shot. Hence, one shot can include different places. As said before the shot implemented in our system undergoes spatial continuity constraints.

Scene The classical meaning of a scene is that of an entity characterized by spatial, temporal and narrative continuity. In the traditional script a scene is always accompanied by two attributes: space and time. For example: scene No.1 - mother and daughter; internal; day. Hence the scene takes place in a fixed environment (a room of the house), is composed of one or more actions, and includes a fixed number of actors. The entering and the exiting of an actor are the edges of the scene. A change of time, for example from day to night, corresponds to a change of scene. Cinema doesn't always apply these spatio-temporal criteria to delimit a scene. For example: two people who are staying in different houses are taken alternately to show what they are doing at the same time. Though this action doesn't all occur in a fixed place it is considered to be one scene in which narrative continuity prevails over spatial continuity. In cinema a scene is composed of many shots but it can also contain one long shot alone. As said before the scene implemented in our system undergoes spatial continuity constraints.

Block. Several scenes can be logically grouped together into a block if they have a common topic. Directors use this method when various themes are interlaced within the story. In our layout the block is an attribute of the scene.

Sequence. A sequence is a group of scenes taken by one or more cameras and connected together by thematic reasons. In our layout a sequence is an attribute of the scene.

Act or Part. The act is the largest composition unit and specifies mainly the temporal progression of events. Cinema calls it Part.

4.2 Simulation

The simulation takes place in the stage window (Figure stage) and is a real time process which produces animation, sound and speech. There is one scene in memory and the user can run it or load whichever other scene of the performance memorized on disk. Users control the simulation through the buttons of the controller which includes play, stop, pause, forward, backward, and time setting. The control buttons and the current time sliders are located just below the stage as they represent the most frequently used functions. Users have the feeling that they are driving what is happening on the stage in a very natural way. The user can either position the timer at the begin of the scene and then play it, or set the time at some position in between and then run a piece of scene, or simply move to time T_i to visualize the static situation of the scene at this moment. Initially the duration of the scene is zero but it increases automatically as long as the user add actions to it.

The current time is a global variable of the interface: each part of it is kept consistent with the current time and is updated immediately when the time is changed. There are many ways to change the time: for example by either writing the exact figure or moving a slider on a percentage scale or pushing down the forward or the backward button. The percentage scale is very useful to obtain a coarse positioning in a scene, instead the backward/forward buttons let users control time locally. Usually these two methods are used in conjunction: first the user moves the slider to a point in the percentage scale, then s/he adjusts the position by pushing the backward/forward button repeatedly. This functionality meets the mental scheme of directors when they want to position (or search for) an action in a scene. They might prefer to manage the timing of the scene approximately: they want to move toward the middle (to 50% of it) or toward the end (say, around 75% of it).

During the simulation users who want to fix and memorize a particular point can push a mark button: a relevant cue might be a word in a speech, a sound or a gesture of an actor. This mark can be used for synchronizing events as described next in the score.

4.3 Score

The score (Figure score) is something more than a timeline: it represents the symbolic description of a scene much like the music score describes music by means of notes. It is composed of two windows: the left one can be scrolled vertically and lists the time varying elements of the scene in a hierarchical structure; each element on the left has a corresponding timeline on the right which contains a sequence of coloured boxes. For example, the *character* group is split into two items: line (i.e. speech) and motion. On the speech track there are as many boxes as the number of phrases that the character pronounces in the scene. The same for the motion track. Each time the user defines a new element, for example a new motion, it appears immediately on the score.

The score has many editing capabilities: users can move the items to change their start time and their durations. By clicking on a box the user can enter the editing mode and change all the parameters of that element. Users can use the marks set up during simulation to align boxes on the timeline.

5. The virtual rehearsal

Traditional animation systems are based on the storyboard method which consists in preparing a number of key scenes which the system interpolates to show the final animation. To build each key scene the animator positions actors on the stage with their appropriate postures. This approach has been inherited from cartoon techniques and is suitable for making cartoons or films on the computer as well: animators use to design the storyboard first, then, when each frame has been clearly defined, they build it.

Even LifeForms (Calvert et al., 1993), which is not a general purpose animation system, has the same approach. This choice is justified by the fact that choreographers need to work at a key frame level for two reasons. The first is that they need to control the details of a posture down to the joints angles values. The second is that they need to coordinate the key frames of two different figures at the same point in time. In these systems the timeline represents the key frames of all the figures. A user sees the key frames of each dancer disposed horizontally and can move a key frame forward or backward to change the way it matches with the other dancers.

We, on the other hand, have defined a composition model which is more suitable for theatrical directors. The building blocks are no longer key frames but actions and events. At the beginning of the staging process directors do not have a clear and detailed idea of the final composition, hence are not able, nor interested, to design key frames, but they want to start placing actions somewhere in the scene and check the effect of setting one action after or before another. Rehearsals are the basic mechanisms through which directors design and evaluate a scene. Thus we have defined a model for staging centred around the concept of *virtual rehearsal* to give them the possibility to do with the computer what they usually do in reality. The rehearsal approach lies on the notions of *stage*, scene *elements*, and *composition*.

5.1 Stage

The stage is the space where users design and visualize a multimedia scene. Directors are in front of the virtual stage as if it were the real one: there are no constraints on the order of the operations they can perform so that they do not have to conceive their story with a sequential approach. They are free to design and rehearse any action in any order. The main attribute of the stage is time and the stage always displays what happens in the scene at the *current time*. Users can move the time forward or backward to visualize a point in the scene they have built before or to find a point where they want to start defining a new action. The scene on the stage is updated in real time when the user makes some editing on scene elements or when the simulation is running.

The user manipulates actors and objects with the mouse, directly onto the stage. When users drag or rotate an actor, gravity keeps its feet on the floor of the stage, unless the user has disabled this constraint to lift the actor to some upward position.

To design a scene or an action users first devise its main elements and then begin to build one element at a time. An action could be composed, for example, of a character, a text, a gesture, and a piece of music. Users can follow a natural approach: they put the actor on the stage, give it a particular behaviour, assign it a phrase to pronounce with a certain tone, and then specify some music to play. Then if the director wants to play the whole action or only part of it, s/he has simply to move the time until s/he finds the starting point and presses the play button. Whenever the director sees something wrong s/he stops the exhibition and makes any modification s/he wants.

Since the stage is the heart of the design process the main user interface functionalities are concentrated around it. The central position is occupied by the graphic window containing the stage, above there are the buttons of the elements that can be inserted into the scene, under the stage there are buttons for controlling the simulation, much like a cockpit.

5.2 Scene elements

A virtual performance is a multimedia system which includes these media: *stage, set (properties), actors (body), movements, speech, text, musics, sounds, lights, and cameras*. These elements are the building blocks of the scene which users have to specify and combine. The system provides non expert users with high level, easy to use functionalities to compose elements into a scene.

To reduce the user's workload the system is geared to re-use and adapt existing models collected into libraries (there are bodies, sets, objects, musics, sounds, and movements libraries). Each class of elements has its own set of editing functionality which allows the user to control how an element is adapted to the current application. Moreover a library can be enlarged by importing existing standard elements or new data built by experienced people using specialistic commercial software. Geometric objects (actors' bodies, sets), can be designed by CAD experts in collaboration with artists to define libraries customized to classes of applications. By default the stage is a gridded floor but the user can load a different stage from a library that contains application dependent environments. The prototype includes some typical theatrical spaces such as a Roman stage and a footbridge stage.

Notice that the functionality of CATS are not the same functionality provided by a geometric modeller nor an all purpose animation system. In fact modelling from scratch either a geometric object or a movement of a human figure, are extremely difficult tasks which require both a lot of time and considerable skill in using complex modelling techniques (managing NURBS or key frames editors). Also human movements cannot be modelled without having both a precise knowledge of the internal structures and timings of movements, and a specific skill in describing it through a computer interface. Even talented choreographers and dancers find it easier to teach and learn movements by body imitation rather than making a detailed description.

Animated elements, being characterized by a start and an end time, appear on their own tracks of the score. They are either library elements or created directly by users. For example a speech is obtained by recording a natural voice; cameras, lights, and generic objects motions are created by the users by setting up a series of key values. An animation element for a camera corresponds to a shot. Objects animations describe the constrained movement of a part of an object such as an opening door or a rising panel of the set.

Designing a character is the most complex task. A character is made up of physical properties (*body*), *movements* and *speech*. Our system use stick models for bodies because line is the graphic primitive which can be rendered fastest and which is common to all real time animation systems not running on parallel machines. Thus an actor's body doesn't have to be a realistic representation of a character but a symbolic representation of it. Stick figures are recommended in real time animation because solid bodies would inevitably have a fake robotic appearance which would divert users' imagination instead of helping it. A stick figure doesn't reduce the quality of the animation, which depends exclusively on the number of joints and on the way they match natural positions. Our system is based on a human skeleton with 44 joints. This is a high number if we consider that most motion capture systems used for professional applications (advertisements, cartoons) detect a lower number of joints, around twelve. Thus the skeleton we provide is geared to assume very sophisticated behaviours.

The system provides some functions which assist users to customize their characters, though it is not a real body editor. It covers a few requirements: make characters distinguishable from each other on the stage and introduce some abnormality in human figures. There are libraries which contain standard bodies for categories of characters such as women or men or court people. On this basis users can introduce a series of modifications. Characters can have different colours, shapes and sizes. Users can simulate a figure taller or smaller than the norm to represent for example a child, remove a limb such as a leg or an arm, substitute a limb with an object (for example a mask instead of the head), change the standard length of limbs to build people with arms longer than normal. This limited number of changes is enough to make a rough design of a character and assign a schematic personality to it.

A movement library contains the building blocks, i.e. parameterized movement patterns, for creating more complex behaviours. A pattern is a sequence of key frames with default parameters values, e.g. start time, duration, starting position, number of repetitions. It is usually a short sequence which the system is able to adapt to the application by replicating it several times in the direction desired. For example, a walk pattern contains one step. If the user draws a path on the stage, the step is reproduced many times following the path from the beginning to the end. Because of the complexity of articulated movements and the limitations of current motion control systems, techniques based on the reuse and adaptation of existing motion data are a state of art trend in human figure animation research. This approach is considered complementary to keyframing, motion capture, and procedural animation. Currently our system contains very general adaptation functionalities, as described in the next section, which relies partially on Bruderlin's (Bruderlin & Williams, 1995). techniques. This system has a very general architecture which is open to include any new algorithm which generates a new movement from library movements.

A library movement can be designed by an expert animator using either traditional key framing editors (LifeForms, Wavefront Explore), or motion capture systems, or off-line dynamic programs. The current movement library contains a number of natural movements built especially for our system with LifeForms.

5.3 Composition

In an animation and multimedia system, composition deals with a large number of aspects some of which can be very complex and cpu consuming. Composition concerns the capability to define the temporal synchronization of all events, as well as the interferences between two actors or between an actor and an object. Other very important points are the way an actor progresses from a movement to the successive and the capability to combine or perturb existing movements to obtain a new behaviour. Notice that a highly realistic multimedia scene requires either sophisticated (and cpu consuming) system controls or a hard work from the user to set up all the details.

In our system, we provide very basic and general composition techniques which can be executed in real time. Through these techniques users can control a) the temporal synchronization of the scene elements, and b) the spatial and temporal positioning of an actor's movements. Directions are given directly by the users though the system is able to assume some autonomous decisions to facilitate the user's work, as described below in this section. Despite the fact that this system doesn't include any knowledge on actors-objects interactions, some actions such as walking on a ramp, rising stairs, sitting on a chair, or carrying an object in hand, are provided in the prototype implementation with some limitations: they are applicable to default size objects.

The primarily synchronization role is played by the simulation process which guarantees that the event E_1 will always start at time T_1 and event E_2 will always start at time T_2 of the scene time. The simulation process is controlled by the scene clock which provides the correct time. At each tick the animation engine interpolates the data and displays the actions which take place in that precise moment. An approach based on time instead of frame rates, is necessary when there are media different from animation to synchronize.

In the following we describe some aspects that users can control and techniques which assist them:

start time. One main objective of users is to set the start time of an action in relation to the rest of the scene. For example, the user might want an actor to start waving its hands after having pronounced some words when the actor reaches a particular position or is pronouncing a particular sound. In a scene there is always a prevailing element which depends on the artistic style. In theatre speech usually prevails, and all actions depends on text lengths, but in other cases another element might prevails such as the movement. In our system users can synchronize the events with the element that they consider to be predominant. The most natural method is to make it directly *on the stage*: the user sets up the first event (speech), then simulates it and interrupts the simulation when the actor has to begin the gesture, finally s/he defines the gesture. But users can also do it at a later point. They run the simulation and insert a mark whenever they find a point where they want to insert a new element. The marks appear in the score and are a guide for aligning events manually on the temporal line.

subjective view. At any moment users can select a subjective view of the scene from one of the many spectator's eyes that they have defined or one of the cameras. The aim of composition is actually to offer an effective representation to the audience whether they are seated at the theatre or in front of the TV. In theatre the way actors are arranged on the stage has to be evaluated with great care especially when the stage doesn't have the traditional location in front of the audience but is placed, for example, in the centre of a room with the audience surrounding it from each side (footbridge stage).

direct animation. Since users can move time back and forth they can display any moment of a scene and manipulate any dynamic attribute of lights, cameras, and objects in that precise moment. If the user changes the intensity of light to create a particular atmosphere, then s/he can observe the effect from the point of view of a spectator. When s/he changes a camera attribute s/he has an immediate feedback of what the camera sees. A direct interaction with elements and their dynamic attributes differs from traditional animation systems where users have to define a great deal of curves to animate something. The number of curves can be very high as it corresponds to the number of DOFs of each element times the number of elements. Our system tries to limit the specification workload through a high level interface which reduces the DOFs. For example, instead of defining three curves for the x,y,x coordinates of the camera position, users move the camera on a horizontal plane and then raise it to the height desired. To assign colour to a light instead of defining three RGB values, users select one colour from a palette and then adjust the intensity.

positioning actors on the stage. When users want to check whether two actors, say A_1 and A_2, are positioned correctly in a certain moment, they can simply stop the simulation and adjust the position of one or both actors by dragging and rotating them with the mouse. This operation however modifies the movement that the actor is performing in that moment and may introduce an unwanted distorting effect. The system assists the user by applying a recovery procedure which depends on the situation. If this is the starting point of a movement the system simply applies to the movement this new start position and direction; for walking, the path will be translated and rotated so that the first control point of the path will match the new start position/orientation of the actor. The problem is more complex when this is a point within a movement because a good movement has an internal shape which should not be manipulated. For example, if the actor is doing a somersault, what happens if the user drags the actor far from the current position or, even worse, turns the actor 180 degrees? The system has various levels for controlling this operation. From a very low level moving upwards in the system layers there are these possibilities: a) the user is completely free to move the actors and accepts the risk of damaging the motion; b) the system warns the users that the current movement might become ugly; c) the system prevents users from making this operation by automatically re-positioning the actor to its original place after the user has released the mouse; d) for walking it introduces a new control point in the path; e) the system doesn't change the shape of the current movement but translates its start/end positions to match the new key point.

inbetween actions. If there is an interval of time between two consecutive animations of an actor (call them M_1 and M_2), the system assumes that the actor will stay in the end position of motion M_1 until M_2 begins (otherwise the actor would disappear from the stage). Of course if the start position of motion M_2 is not the end position of motion M_1, there will be a spatial discontinuity when the simulation runs. This inconsistency can be handled either by the system or by the user depending on the value of the spatial continuity attribute described next. Also the creation of a proper sequence of movements which represent a credible action depends on users. Let us make two examples. Suppose actor A_1 who is lying on the floor has to sit on a chair. The correct sequence of movements that a user should specify is: *stand up, sit down..* Actor A_2 has to walk to a point and sit on a chair that is turned 90 degs with respect to its actual orientation. The correct sequence of movements that a user should specify is: *walk, turn, sit down.* A more advanced type of control is described in the next section.

spatial continuity. By default a new movement is affected by *spatial continuity* with respect to the previous movement. This means that motion M_2 starts from the end position of motion M_1 which precedes it in time. The spatial continuity is an attribute of each movement and determines whether it is linked or not to the movement which precedes it in time. Suppose that the user changes the start time of movement M_1, then it will automatically change also its start position to match the end position of the movement that now precedes it in time. Spatial continuity is kept by the system unless the user switches it off. The user can also force the start point of a motion by writing the coordinates directly in the motion tab or by translating the actor with the mouse on the stage; this user action automatically turns the spatial continuity attribute to off.

5.3.1 Advanced composition features

This system gives the possibility to combine two movements taken from a library and create a more complex movement. For example, the user might want an actor who has been walking for a while to start waving his/her hand. Users do not have to make any special operation because the system automatically recognizes when two movements overlap in time and applies the right procedures. Users only have to specify the actions in the temporal order they want: first users define a walk then, after a few steps, they assign a gesture to the actor. The user can check and modify the overlapping times on the score where the boxes of the two movements appear on two different parallel tracks. The algorithm for merging parallel movements is derived from the techniques described by Bruderlin (Bruderlin & Williams, 1995). A similar approach is applied to smooth the transition between two consecutive movements of an actor. These advanced functionality are still under development and testing.

6. Conclusions

The system described is still under development and its functionality are under evaluation and consolidation. There is no way to mathematically quantify the results since the system features are highly subjective. Because of this, a variety of people have been consulted on what they think of it. At all stages it is tested by both non expert and expert users. External users are very useful because they bring a naive point of view which can be very creative. New requirements concern either the access, at the user interface level, to new information, which are available in the internal data structures, or the creation of new data and new algorithms for their management in the animation system. This latter condition implies a great deal of work to extend the internal data structures or add new ones. On the other hand, all evaluators have testified their great interest and expectations for the overall approach and the user staging model and their contribution aimed to refine and complete the system functionality, not to overturn the basic concepts so far defined and implemented.

To improve the capability of the system to create valuable scenes, a great effort is being undertaking to find a useful classification of human being natural movements and create a movements library. Such a work cannot be done without the competence and strict collaboration of an expert choreographer such as Roberto Castello. Differently from other ongoing researches that aim at building libraries for ballets, we concentrate on *natural movements*, a not yet standardized problem. The result of this work will provide not only a library of movements but, most of all, a composition language for movements.

Acknowledgements

We would like thank all those directors and writers from the theatre television, and cinema, along with the set designers and actors, who disclosed the secrets of their work and contributed to designing this system. A special acknowledgement to the theatre director Gilberto Tofano who put his faith in the capacity of the computer, and the choreographer Roberto Castello whose knowledge of the behaviour of the human body together with his skill in computer animation was fundamental for designing character animation features.

We acknowledge the European Commission for funding this research under Esprit Project No. 20340.

7. References

Barenholtz, J., Wolofsky, Z., Ganapathy, I., Calvert, T. W., & O'Hara, P. (1977). Computer Interpretation of Dance Notation. In S. Lusignan & J. S. North (Eds.), Computing in the Humanities, (pp. 235-240). Waterloo: University of Waterloo Press.

Benesh, R. (1956). An Introduction to Benesh Dance Notation. London: A. and C. Black.

Breiteneder, C., Gibbs, S., W. Heiden, W., Kaul, M., Jonas, K., Schnepf, U., & Steinberg, D. (1996,). ATM Virtual Studio Services. Paper presented at the 6th International Workshop on Network and Operating

Systems Support for Digital Audio and Video (NOSSDAV 96).

Bruderlin, A., & Williams, L. (1995,). Motion Signal Processing. Paper presented at the Computer Graphics Proceedings.

Calvert, & Chapman. (1978,). Notation of Movement with Computer Assistance. Paper presented at the ACM Annual Conference.

Calvert, T., Bruderlin, A., Dill, J., Schiphorst, T., & Welman, C. (1993). Desktop Animation of Multiple Human Figures. IEEE Computer Graphics & Applications(May), 18-22.

Calvert, T. W., Bruderlin, A., Mah, S., Schiphorst, T., & Welman, C. (1993,). The Evolution of an Interface for Choreographers. Paper presented at the INTERCHI'93.

Earnshaw, R. A., Gigante, M. A., & Jones, H. (Eds.). (1993). Virtual Reality Systems. London: Academic Press.

Eshkol, N., & Wachmann, A. (1958). Movement notation: Weidenfeld and Nicolson, London.

Feldenkrais, M., & Eshkol, N. 50 Lessons : Research Center for Movement Notation, Faculty of Fine Arts, Tel Aviv University.

Gibbs, S. (1996). Interaction in the virtual studio. Computer graphics, Nov. 96.

Hutchinson. (1970). Labanotation. New York: Theater Arts Books.

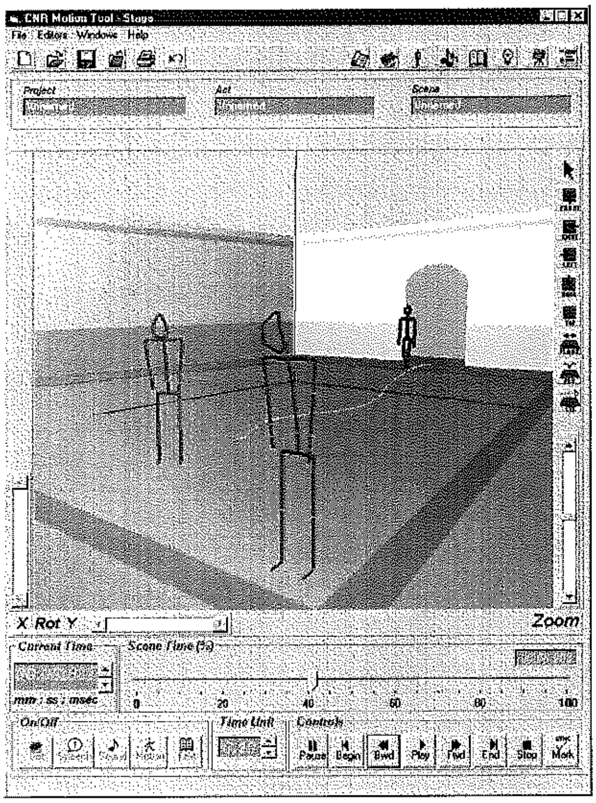
Palamidese, P., Tofano, G., & Scapellato, G. (1996). CATS: TV, Cinema, Theatre, Drama Schools Requirements (CNUCE Report C96-25).

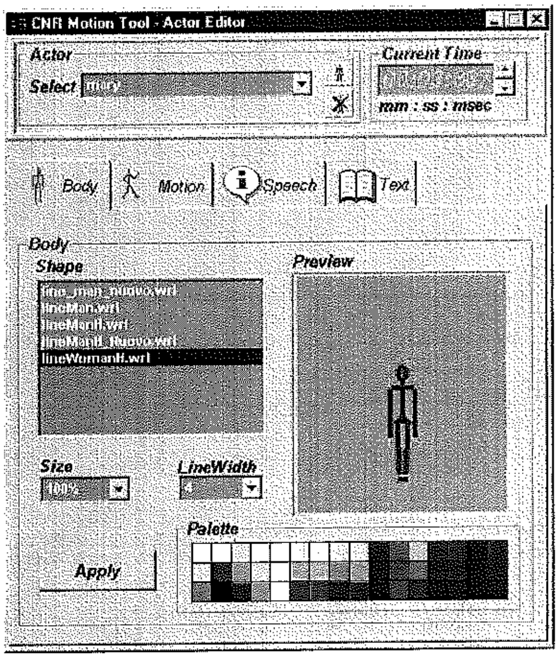
Sealey. (1979). Recording Movement in Labanotation on Computer (User Manual): University of Iowa.

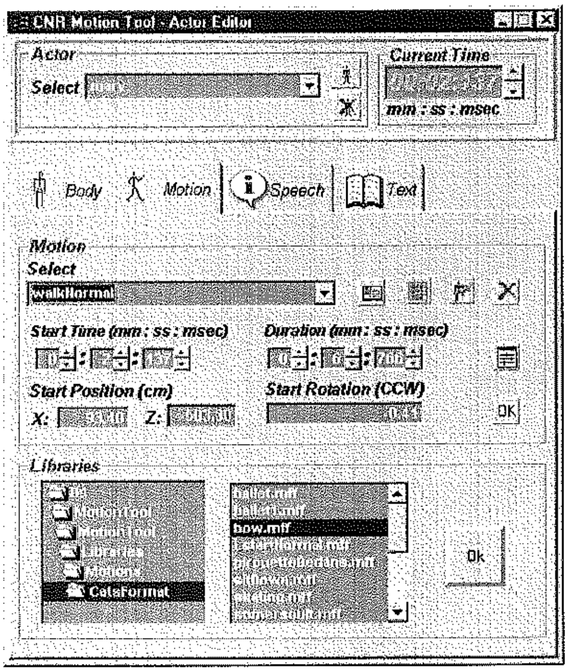
Smoliar, & Weber (Eds.). (1977). Using Computer for a Semantic Representation of Labanotation. Waterloo, Ont.: University of Waterloo Press.

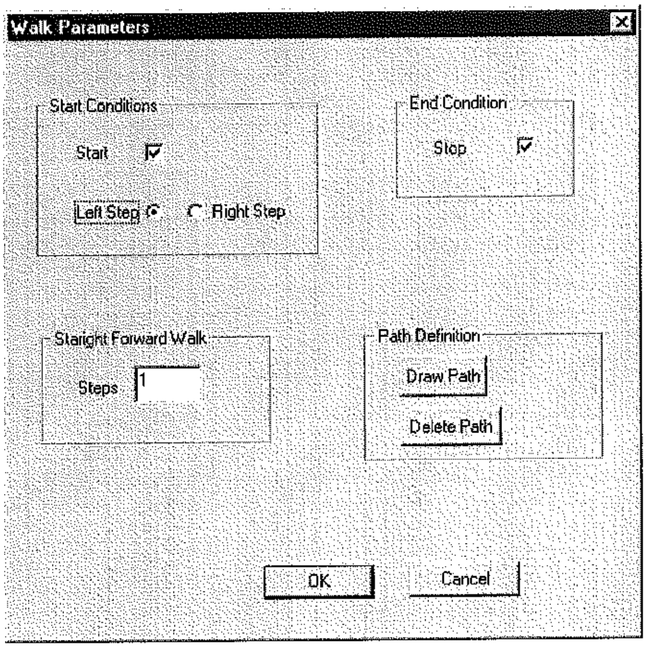
Thalmann, D. (1993). Using Virtual Reality Techniques in the Animation Process. In R. A. Earnshaw, M. A. Gigante, & H. Jones (Eds.), Virtual Reality Systems, (pp. 143-159): Academic Press Ltd.

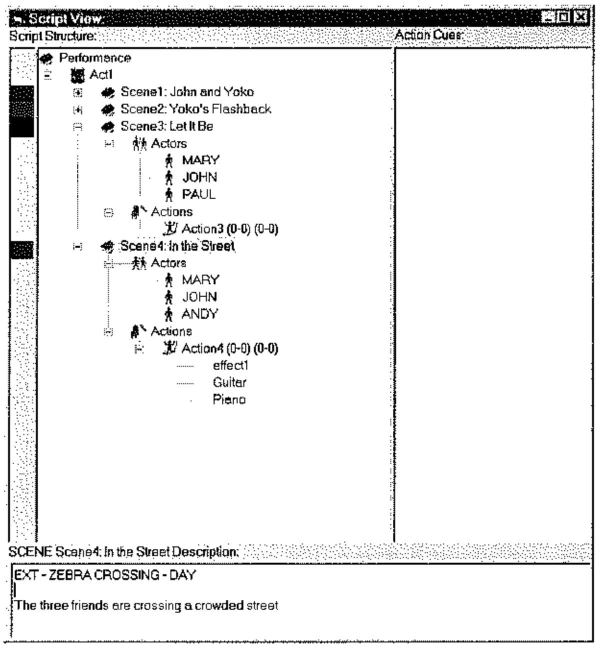
Wolofsky. (1974). Computer Interpretation of Selected Labanotation Commands. Unpublished MS, Simon Fraser University Kinesiology dept.

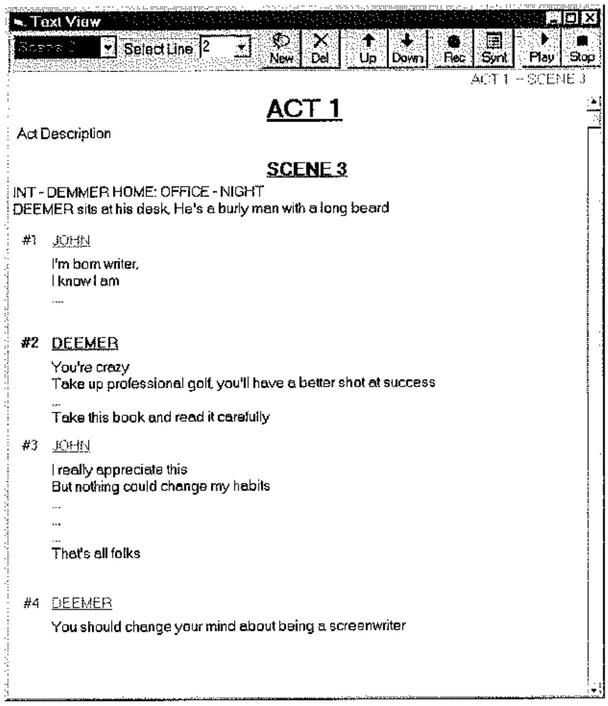

CNR Motion Tool - Stage
File Editors Windows Help
Project
Act
Scene
X Rot Y
Zoom
Current Time
Scene Time (%)
mm : ss : msec
0
20
40
60
80
100
On/Off
Time Unit
Controls
Pause
Begin
Bwd
Play
Fwd
End
Stop
Mark

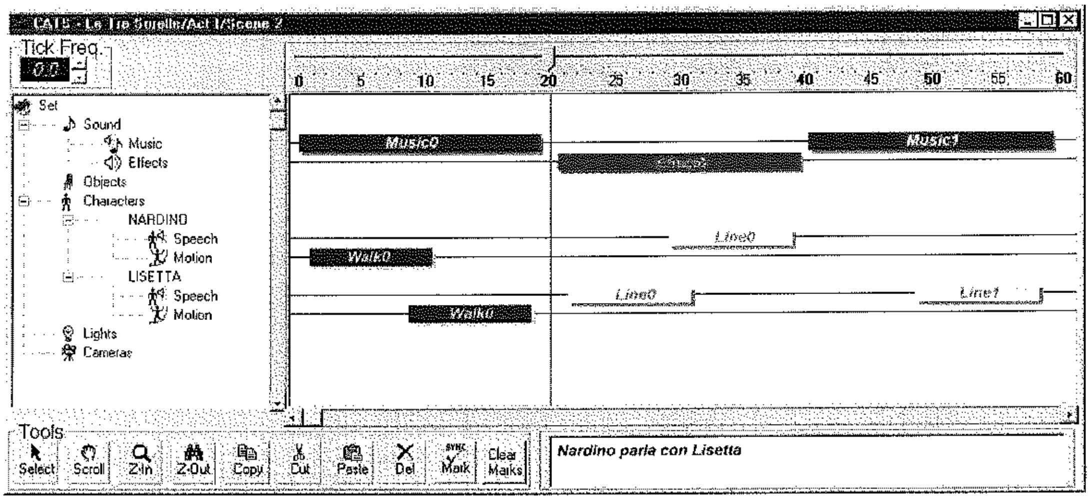

CNR Motion Tool - Actor Editor
Actor
Select
mary
Current Time
mm : ss : msec
Body
Motion
Speech
Text
Body
Shape
Preview
line_man_nuovo.wrl
lineMan.wrl
lineManH.wrl
lineManH_nuovo.wrl
lineWomanH.wrl
Size
LineWidth
100%
4
Apply
Palette


CNR Motion Tool - Actor Editor
Actor
Select
mary
Current Time
mm : ss : msec
Body
Motion
Speech
Text
Motion
Select
walkNormal
Start Time (mm : ss : msec)
Duration (mm : ss : msec)
Start Position (cm)
Start Rotation (CCW)
X:
Z:
OK
Libraries
MotionTool
Libraries
Motions
CatsFormat
ballet.mtf
bow.mtf
sitdown.mtf
skating.mtf
Ok


Walk Parameters
Start Conditions
Start
Left Step
Right Step
End Condition
Stop
Starght Forward Walk
Steps
1
Path Definition
Draw Path
Delete Path
OK
Cancel


Script View
Script Structure:
Action Cues:
Performance
Act1
Scene1: John and Yoko
Scene2: Yoko's Flashback
Scene3: Let It Be
Actors
MARY
JOHN
PAUL
Actions
Action3 (0-0) (0-0)
Scene4: In the Street
Actors
MARY
JOHN
ANDY
Actions
Action4 (0-0) (0-0)
effect1
Guitar
Piano
SCENE Scene4: In the Street Description:
EXT - ZEBRA CROSSING - DAY
The three friends are crossing a crowded street


Text View
Select Line
New
Del
Up
Down
Rec
Synt
Play
Stop
ACT 1 -- SCENE 3
ACT 1
Act Description
SCENE 3
INT - DEMMER HOME: OFFICE - NIGHT
DEEMER sits at his desk. He's a burly man with a long beard
#1 JOHN
I'm born writer,
I know I am
#2 DEEMER
You're crazy
Take up professional golf, you'll have a better shot at success
Take this book and read it carefully
#3 JOHN
I really appreciate this
But nothing could change my habits
That's all folks
#4 DEEMER
You should change your mind about being a screenwriter


CATS - Le Tre Sorelle/Act 1/Scene 2
Tick Freq.
0 5 10 15 20 25 30 35 40 45 50 55 60
Set
Sound
Music
Effects
Objects
Characters
NARDINO
Speech
Motion
LISETTA
Speech
Motion
Lights
Cameras
Music0
Music1
Line0
Walk0
Line0
Line1
Walk0
Tools
Select
Scroll
Z-In
Z-Out
Copy
Cut
Paste
Del
Mark
Clear Marks
Nardino parla con Lisetta