Interaction in Virtual Worlds: Application to Music Performers

J. Esmerado, F. Vexo, D. Thalmann
Computer Graphics Lab
Swiss Federal Institute of Technology
EPFL-LIG 1015 Lausanne

Abstract

We present a model for the representation of the interactions between virtual human figures and virtual objects in 3D virtual scenes. These interactions can depend on externally provided information not being limited in their complexity. The representation model for virtual humans is modular and provides tools for representation of the interaction know-how pre-requisite. Timing constraints are relevant in this model and concurrency and synchronism are used to insure the adequacy of the resulting animations of virtual humans and associated interacted objects. Resulting gestures are derived from an adapted application of Inverse Kinematics methods.

Keywords: Virtual Scenes, Virtual Humans, Virtual Objects, Interaction, Inverse Kinematics, Musician Simulation.

1. Introduction

Populated virtual environments have been a wide research topic in recent years. Among the possible trends within this field there is the quest for realism and believability. In virtual environments, believability depends on several factors such as an accurate rendering of the visual properties of the scene constituents. Believability can depend also on the accuracy and naturalness of the possible events and actions taking place in the virtual scene. Events may result from actions produced by active scene constituents, such as autonomous virtual humans (AVH) and the accuracy of these actions and associated events depends on the adopted underlying model for the virtual scene.

One of the expected important types of actions is of the interaction type. In this work we are especially interested in the representation of interactions between autonomous virtual humans and other scene constituents named *virtual objects*. Virtual objects can exhibit several levels of morphological and motion inducing methods nature complexity. Some virtual objects can exhibit self-animation

properties; other virtual objects require virtual human guidance to perform in an adequate, believable manner. The latter corresponds to the case of virtual human/virtual object interaction based on information about the object and its standard use.
In order to handle the variety of elements that can be part of a virtual scene and their relationships in a comprehensive and semantically adequate manner, an integrating scene content model is necessary.

This paper presents a proposal of a comprehensive model for representing the content of virtual scenes. This model is meant to facilitate capturing the relationships and semantics of virtual scene contents and to provide a framework for the effective representation of interactions between elements in the virtual scene. This is model is based on an object-based approach to enable reaching any level of particularity for a given scene constituent.

2. Context

2.1 Proposed 3D Scene Model

In a virtual scene, it is possible to distinguish the constituents, (henceforth referred to as *elements*), actions and events. Elements can be non-animated (or *static*), or animated. Elements can also correspond to virtual objects (including a particular case, the background images or *décor*) and virtual humans. Virtual Humans can exhibit either autonomous behavior or be 3D representatives of real world users (real-user avatars). Virtual Objects can range from simple non-animated, non-deformable types of object to deformable, self-animated and reactive ones.
In virtual scenes, events can derive essentially from actions performed by active elements in the scene and is typically associated with the beginning and ending of some relevant activity [9]. Specially interesting for an external observer is witnessing realistically simulated interactions in virtual scenes. The level of complexity of the interaction will be able to draw a corresponding level of interest of an observer [3][6]. For this reason, one of the principal objectives of our model is to facilitate the representation of sophisticated interactions in virtual scenes taking place in an automatic way.
In our model, we are interested in representing mainly, but not only the visual properties of the scene elements. This way, visual properties of elements are identified as their *shape* and a particular shape can be associated to an element for rendering purposes.

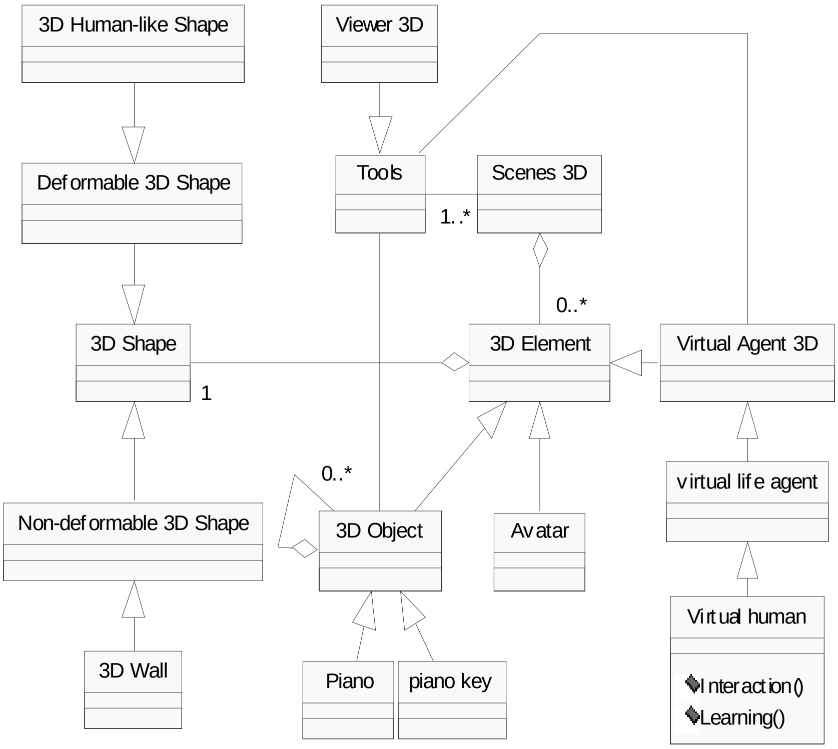

Figure 1: 3D Scene Content Model

The representation of Virtual Objects must encompass all the range of their possible levels of variety and complexity along with common properties such as properties related to their morphology. For reaching an acceptable degree of realism the object representation should contain representations of forms of behavior that relate to the physical properties of that object. This can become particularly useful when events such as interactions between the object and other scene elements take place. This way, in many cases a virtual object should be capable of performing self-animation, that is, to generate its own forms of reaction from its internal representation.

The internal representation of self-animation may need to accept input from the surrounding scene. Controlled self-animation corresponds to a situation where another scene entity (generally a virtual human) is commanding or at least guiding the object's animation. In our model it is considered as too costly for rendering time to base interactions representation strictly on the simulation of physical contact. The consequence is that commanding a virtual object corresponds to starting and ending each of the object's animations, being the object's internal animation representation that takes charge in-between. An autonomous animation of an object corresponds in short to a behavior from the object. An object can be composed from several sub-objects or parts, forming a more complex whole. Each of those components is an object in its own right that can possess a similar set of attributes to the main encapsulating object.

Figure 1 shows an example of the integration of elements in the scene their explicit associated shape properties and the possibility of defining objects from other objects.

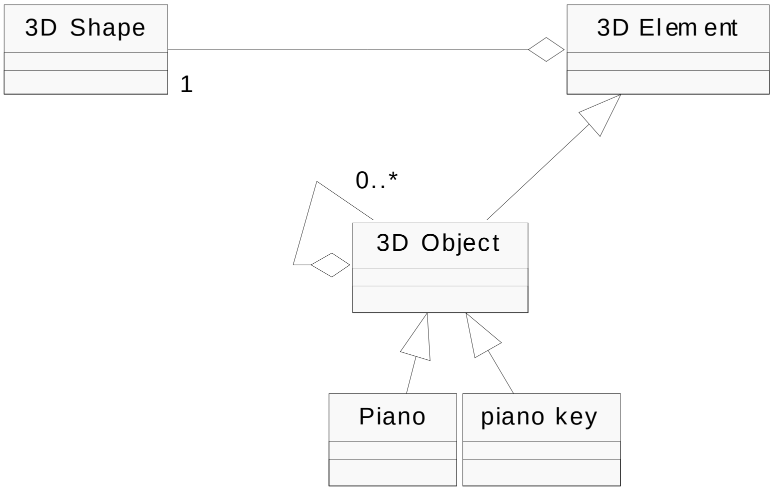

Figure 2: Virtual Object Model

A virtual human model must contain the necessary elements allowing it to be visualized in a realistic manner while possessing the characteristics enabling an informed, varied and autonomous type of interaction with its surroundings. This latter aspect entails the need to associate to the virtual human the necessary resources to allow for forms of perception, interpretation, reasoning and gesture generation adapted to the situations generated in virtual scenes. We call the latter intellectual attributes. These may encompass a vast set of properties or criteria. For this reason we will narrow down the discussion of mental properties representation to those attributes that are relevant for the representation of virtual human interaction skills.

For representing skills it becomes necessary to represent knowledge [8]. Knowledge is generally applied using also information data and all those elements must be stored or *memorized*. Limited duration memory data is considered as part of a

Volatile Memory whereas lifelong data is stored in a Permanent Memory region. All information that must become part of a virtual human's patrimony is a permanent memory item. That is for example the case of skills. In Volatile Memory items like recollections of encountered objects in the scene, objects currently being carried, etc. These items are associated to a maximal duration and can also be discarded when they stop being relevant.

Interaction skills are normally associated to one or more objects. To facilitate representing the need of an object a dedicated *known objects list* is proposed in order to quickly determine whether an encountered object permits the impending execution of one of the virtual human's skills. By crosschecking the encountered object's type identifier with the *known objects list* it is possible to determine a list of associated skills and by matching with all available skill preconditions to see if the skill execution can proceed.

For ease of representation of the list of objects being carried at a given time by a virtual human a memory *inventory* list is also introduced.

2.1.1 Skills Representation

Skills Representation

{

Executable skill types to choose from:

- Identification Skill Model
- Independent Basic Skills (Walk, Dance, Music, etc.)
- Procedural Skills (linked to Virtual Object)

Additional possible characterization data objects per skill:

- Skill Level
- Time Influence on Skill Level (law of evolution)
- Skills Dependence on VO
- Skill Dependence on other Skills

}

Among virtual object-related skills, some simplifying modeling choices are adopted:

- The levels of necessity of a given object can increase/decrease over time account for the increased longing for performing a particular Skill.
- The Pre-processing of the scene is required for identification of the objects present that are concerned by skills. This mechanism can be limited to a perimeter around the Autonomous Virtual Human (AVH).
- A filtering mechanism discards the useless objects (those with no skill associated for the current AVH).

A *pseudo-code* for Interesting *Object Selection Mechanism* is as follows:

```
if ( FoundObject IN ListofSkillRequiredObjects ) { //
        FoundObject -> LocatedObjectList
        if ( FoundObject IN NeedList ) { // Especially Necessary Object
                if ( FoundObject == AVAILABLE ) { // Not taken by other
interacting VH
                        GetReferencePositionAndPostureToAdopt()
                        MarkObjectAsTakenByCurentVH( FoundObject )
                        StartInteractionSkill()
                }
                else { // Object taken...
                        while (NOT tooLong() ) {
                                if ( OtherObjectSameTypeInKnownLocation() )
                                {
                                        // There's still some hope...
                                        GotoSameTypeObjectOldLocation()
                                        if ( StillThere() AND AVAILABLE ) {
                                                StartFromTheBeginning()
                                        }
                                else {
                                        ContinueSearchfor Objects()
                                }
                        }
                        QuitCurrentlyActivatedSkill() // Liberates resources
associated to the current skill
                }
        }
}
```

A *Need* List allows the virtual human to give preference to certain types of objects among those that were identified in its neighborhood. The intersection of this list with the available objects allows for possible Need-list entry satisfaction. The object in question must be marked as *free* to become associated to a virtual human and must also be marked as *portable* for the current AVH in order to be able to be picked-up by it. The complete representation of a Need concept would entail abstract needs derived from rules like 'people must wear a helmet before going into a construction site'. A skill that requires a missing object cannot be activated until that object is localized and *captured* (associated to the virtual human). It can happen that a skill requires more than one object and this should associate to a single missing object a very high priority during search and identification phases.

There are two important notions: *Interesting* object, which corresponds to objects within the set of skill-related objects in the Autonomous Virtual Human's current skill list, and the notion of *Especially Necessary* object, which corresponds to the contents of a special list within the Autonomous Virtual Human model, representing an object need to be fulfilled as soon as possible. This avoids having to manage a representation of a growth of need over time or some other equivalent method.

2.1.2 General approach for modeling virtual object interaction skills

As depicted from the diagram in Figure 3, our general approach to skill modeling consists firstly of taking all the necessary data from adequate sources such as information providing objects in the scene and converting that data into a skill-usable data format if required. Once all the necessary data (including knowledge data) is available in a convenient format, interaction skill data generation can take place using adequate reasoning mechanisms.

The results may consist mainly on animation data and also the required results or specific effects on the interacted object. As depicted Figure 3, the interaction module feeds a synchronization mechanism that enforces the simultaneity of the resulting animations on the Autonomous Virtual Human embodiment and also on the interacted Virtual Object, along with possible interaction results or products such as sound generation, lighting effects, etc.

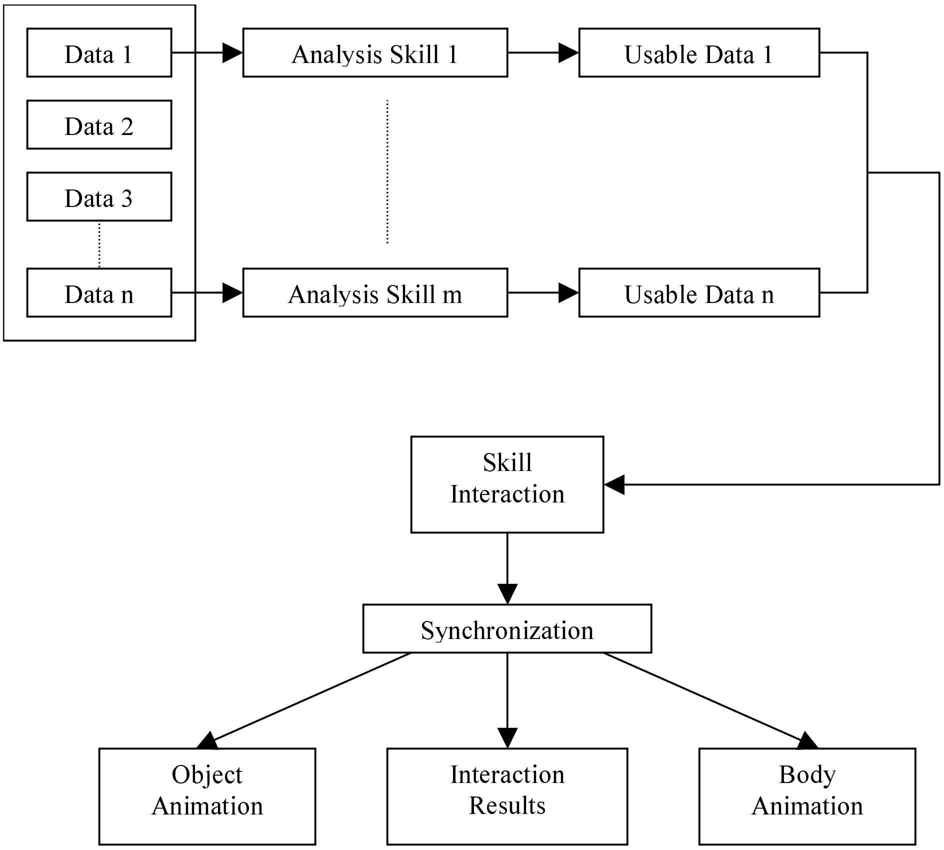

Figure 3: General Interaction Model

3. Application Example: Virtual Pianist

As a demonstration example for the usage of our proposed model we chose the case of a virtual pianist. This demonstration is chosen as an illustration of what a complex interaction case may involve and of how the proposed model tools are used in order to solve the interaction simulation problem. In this case, the virtual scene must contain at least a Piano virtual object a Pianist autonomous virtual human and a virtual scorebook in order to provide the necessary music input. No music score data is initially associated to the virtual human.

The particular demands posed by this type of interaction include:

- Pianist must act within a very strict timeline due to music dependence. Music is essentially a sound-time form of activity.
- The contact surface of the key (piano mobile component) is very restricted. The finger as arm extremity should accompany the motion of the key
- The speed associated to those motions should reflect the produced sound volume.

The interaction between finger and piano key is dependent from music data input. It has to be automatically decided by a specialized Expertise. It is also dependent on the availability of a carried object containing music data, a music score object.

3.1 Permanent Memory Content

Interaction skills always obey a general pattern described as follows:

- Approach the Virtual Human to the Virtual Object to be interacted and place the Virtual Human embodiment and/or the Virtual Object at a suitable distance according to the skill specification.
- Execute the skill interaction with the virtual object.
- Restore the previous Virtual Object state and abandon its immediate surroundings.

For a piano playing interaction to take place in a virtual scene the virtual human must possess dedicated skills as described in the following.

3.1.1 Piano Playing Skill Execution Model

The piano playing skill general model normally follows this sequence:

- Locate piano virtual instrument
- Check if music input is already available for the Virtual Human
- If there is music input available proceed as follows:
- Place music Input Object if it exists and is necessary
- Make Virtual Human install itself at a suitable position in front of the instrument. Hands and arms float closely over the keyboard without possibility of contact
- Music Playing Skill Execution phase
- Remove music Input Object if it exists
- Virtual Human abandons the location in front of the instrument

3.2 Music Playing Skill Execution Phase

The piano skill Execution model generally obeys the following procedural pattern:

- Transform Note Input into structured note data
- Generation of a music interpretation
- Hand/Finger attribution to the note
- Schedule time for note execution
- Feed time and note to the instrument model
- Feed finger and time to the Virtual Human animation unit
- Generate animation for Virtual Pianist
- Generate animation for Virtual Piano
- Check end of execution for each time-scheduled note set for both Virtual Pianist and Virtual Piano

3.2.1 Fingering Attribution Skill

This capability is a skill that does not refer to interaction directly. It corresponds to the execution of an abstract model according to which it is possible to attribute fingers to keys given, at least notes and rhythm.
General models for tackling this problem have been proposed. Their tendency is to rely on mathematics [7]. Our approach seeks to approach near real-time performance, as in real life sight-reading. It is a rule-based system approach enriched with meta-rules, heuristics and search mechanisms. The method consists of applying pre-defined fingering knowledge as much as possible. This technique corresponds to isolating parcels of the stream of notes that match a partial predefined solution. Frequently, there may be more than one possible solution. In a rule-based approach, the preferred solutions are those that solve for the largest number of notes. Heuristics are applied for guiding those choices especially concerned with frontier conditions between partial solutions to help disambiguate conflicts and provide a satisfactory overall solution for each case [5][11][8].

The used rules are about permitted fingering sequences, exceptional situations and physical *convenience* rules. In music, some special 'rules' apply to certain pre-defined situations. When piano music is the focus, several identifiable higher-level musical elements possess their own set of possible fingerings and thus these elements should be clearly identified beforehand. This identification allows the use of rules that mention those consecrated musical units, in general harmonic elements. Element examples include chords, the arpeggios, tremolos, octaves, octave scales, broken chords, harmonic intervals and scales. There's also general knowledge on how certain harmonic interval successions in fact can also be seen as broken chords.

3.3 Scorebook and Music Data Representation

A virtual human interacting with a virtual piano object is based on musical input. To be able to provide different music pieces at different points in the scene without the need to store all that information in the virtual human itself we introduce the virtual scorebook in the scene. A pianist virtual human can then use the musical information contained in this object to guide its interaction with an available virtual piano.

If a scorebook is present in a suitable position with respect to the virtual human and the piano, or if a scorebook is part of the virtual human's inventory then a link is created between this object and the virtual human so that the processed music data can be transferred.

The model for the scorebook virtual object derives from the general object representation model and more specifically contains references both to a MIDI source data file and to the proprietary format used for the animation processes.

Concerning music data, the requirements of an automatic music-based instrument plus virtual human animation system are not directly met by the MIDI specification as defined. The reasons that dictated the adoption of an alternative representation for the music data are as follows:

- In order to be able to implement a reasoning scheme based on music data there is a great convenience of being able to structure such data thus forming when possible higher level data structures that allow a more comprehensive vision about the music properties. If we only dispose of raw one-by-one note data there is a strong limitation as to the variety of rules one can write about music content. In other words structuring music data is the key to dispose of an effective music analysis tool.
- In order to be effective, at a given point in time the animation system should promptly know which and how many notes are active in a structured way, independently of MIDI internal file structure considerations such as number of intervening tracks and the partitioning of notes among those tracks.
- The needs of an automatic choice of fingerings (attribution of hands and fingers to the set of notes being play at each moment) also imposes that at each instant in time we should be able to take a theoretical snapshot of the keyboard state and that we can do the same for any number of instants preceding or coming after the current instant. This is due to the input requirements of the expert knowledge approach typically associated to this type of choice making.
- The body motions generated by a real virtuoso pianist can reach high speeds, especially at the finger level. To be able to emulate a real-time interaction it is necessary to limit the minimal duration of a musical event, otherwise the system would be forced to frequently discard motions that cannot be rendered during a given time lapse in order to keep up with a given musical pace. Since in that case musical notes would still be

rendered, it would be senseless to be forced to discard frequently the corresponding animation.

Our approach for structuring musical data consists of the following fundamental steps:
Notes are collected and their durations noted, along with their starting times, independently of their MIDI track origin.
We estimate the minimal duration of a single note.
The previous choice allows also for a representation of the music content in a fixed rate basis. The minimal time duration is used to define a ”music frame” duration time unit. This way all notes can be represented over time in an integrated manner as multiples of the specified minimal time duration.
This representation allows representing the processed MIDI content as entries into a table where each entry corresponds to one single minimal time duration. The played notes can be represented as part of as many consecutive table entries as necessary to represent their total duration. The representation as a table allows to immediately knowing the past present and future of the notes being used and therefore is a valuable tool for the automatic generation of fingerings.

4. Implementation

Currently an implementation of a Virtual Pianist playing on a Virtual Piano using data from a Virtual Score book exists based on the proposed Scene Content Model. This implementation is meant to provide a more general Music Simulation Platform integrating any number of musicians playing assorted instruments. Languages C/C++ are used both on Unix and Windows platforms. On Windows, proprietary VHD++ platform development tools are used essentially for rendering.

The visual rendering is issued through a separate lightweight process, which controls also accessory aspects like viewpoint positioning and the *camera motion*. This allows a convenient positioning of the observer. The sound rendering is not sensitive to this positioning yet, though.
The synchronized animations are shown to the observer through the visual rendering process. The animation of the scene elements is done through synchronized lightweight processes (Posix threads) that react on a quarter of a second basis in time for the rendering process to do its image updates. Also synchronized with the animation of the scene elements (piano and pianist) the piano sound rendering thread operates based on the concept of a minimal duration for a single note. Every single note has to be represented as a multiple of the minimal note duration.

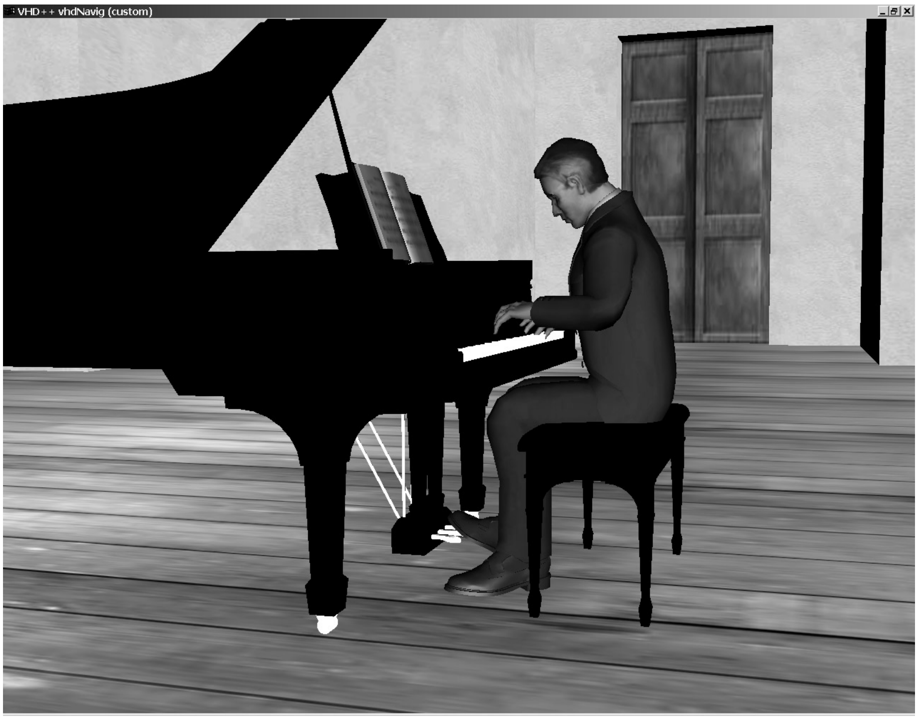

Figure 4:Virtual pianist interacting with virtual piano.

In our implementation, music data usage is prepared before any animation takes place. This can be viewed as a *learning* phase, or pre-processing.

The piano learning process is intimately related to the musical piece being played, as it would be in a real pianist case. In our case learning means learning the gestures for the playing of a specific music piece. To this end a piano-learning routine is centered on one function where the necessary know-how to generate the standard representation of the body articulation angles for each animation frame is represented. This corresponds to a modular representation for the know-how of a specialist.

The main implementation aspects covered by the pianist-related know-how function are:

- Collecting of relevant note information (current, past and future)
- Attribution of hands and fingers to every note in the current time slot (as dictated by the internal music table representation entry) and
- Generation of the gestures corresponding to the playing of the target notes using the attributed fingers taking into account as many positioning factors as the internal skill representation of the pianist allows to.

The first activity, finger attribution to keys, was addressed by using a system based on rules. This system may be as comprehensive as required by the particular application demands, thus providing an ever-extendable refinement, which can provide multiple solutions, that can be ordered and selected using heuristic criteria.
Our music representation structure facilitates this type of analysis for rules condition matching, contrary to the original MIDI format given the fact that all the content is directly available. For each instant in time, the solutions found depend on the current set of notes to be played but should also depend on recent past notes and also future notes to be played in order to generate the most natural possible gestures/solutions. Obviously, there is normally no unique ideal solution to this type of problem. In our implementation we only strive to produce reasonably well-formed solutions, since in general there is no unique absolutely "correct" solution to the fingering problem, anyway. There are no special duration restrictions to this phase since the resulting finger attributions are generated "off-animation" and stored away in the knowledge memory of the associated virtual human for later usage.
The second, gesture generation, uses inverse kinematics techniques to generate the final gestures associated to the placement of the fingers on the desired (virtual) keys, these being in their pressed-down position [1]. The inverse kinematics algorithm conversion is normally insured by the specification of a number of controlling points (end-effectors) situated at the fingertips and wrists of the virtual human pianist [2][4][10]. The resulting gestures are combined with in-between gestures and interpolated so that a smooth gesture sequence and transitions results [12][6].

The learning phase is meant to produce adequate gestures from the available input and corresponds to a modular part that is replaceable. This modularity property is one of the important points of the object-oriented design of our model. The criterion to abide by is that the output has to be in any case a sequence of gestures adapted to the external musical input. Once the off-animation learning process has produced the set of gestures that correspond to the musical data the animation activity can take place when needed.

The animation phase deals with the interaction simulation. Both the acting element (the virtual human pianist) and the virtual object being acted upon have to be animated in a coherent manner with one another. If the object is composed of observable moveable parts this aspect becomes even more obvious from an observer point of view. We realized that a more physical-based approach involving real time collision detection was not fast or reliable enough at this time of writing if our aim includes also preserving the intelligibility of the music sound rendering. According to the proposed model, we produce the synchronized, independent animations of the player's embodiment and the musical instrument, both synchronized with the output of music sound. Each of these three activities is launched in parallel, their key events being synchronized with one another.

5. Conclusions

We proposed a model for the representation of the content of virtual 3D scenes populated by virtual humans and containing objects that may exhibit complex features.

In this work we have proposed a model for representing the contents of virtual scenes aimed at facilitating the representation of the interactions between virtual humans and virtual objects in those scenes. This model proposal results from the need of having an integrated approach to represent 3D scene content extensible enough to encompass future developments and additions while already providing a frame work for integrating a new virtual human - virtual object skilled interaction modeling. These skilled virtual humans interactions can be modeled taking time into account. Time dependence is a key aspect for the correct representation of many interaction activities especially when both interacting parties exhibit visual animation behaviors and this why we pay special attention to finding solutions to tackle this particular problem.

In our demonstration application we have one virtual human agent (a virtual pianist) that is capable of interacting (playing) with one virtual object (a virtual piano) in an autonomous way. As sole input to the interaction there's yet another object (a virtual scorebook). The scorebook role in the scene is to stock musical data that a virtual pianist can extract and use as input for an educated learning phase and subsequent playing phase. This musical input data from the scorebook is also used for generating the piano animation and corresponding sound production.
One of the most relevant characteristics of the piano-playing problem is the synchronization aspect of music playing animation.
In order to model the synchronization mechanism it is necessary that the data that serves as a basis for the control assumes a suitable format. In our demonstration case the musical input provided is associated with the scorebook scene component. Originally this data comes in MIDI format. However, MIDI format, though structured, does not directly provide a directly usable data source format adapted to our needs: In fact, our needs include the convenient ability to dispose of a list of all the events that take place at the same time and to know exactly what that time is. MIDI does not directly provide data in this format since it is a sequential description of events that may be scattered across multiple independent tracks.

The learning procedure is meant to produce adequate gestures from the available input and corresponds to a modular part that is replaceable. This modularity property is one of the important points of the object-oriented design of our model. The criterion to abide by is that the output has to be in any case a sequence of gestures adapted to the external musical input. Once the off-animation learning process has produced the set of gestures that correspond to the musical data the animation activity can take place when needed. The use of interpolation techniques can further smooth resulting gestures out.

The demonstration application validates the usefulness of the proposed model for interactions in virtual scenes. By introducing improved expert interaction know-how modules the resulting interactions can be further improved in gesture variety and believability.

References

Baerlocher P., Boulic R., *Task-Priority Formulations for the Kinematic Control of Highly Redundant articulated Structures*, In Proceedings of IROS, Victoria, Canada, 323-329, 1998.

Baerlocher P., *Inverse Kinematics Techniques of the Interactive Posture Control of Articulated Figures*, PhD Thesis, Swiss Federal Institute of Technology at Lausanne (EPFL), Lausanne Switzerland, 2001.

Becheiraz P., *Un Modele Comportamental et Emotionnel pour l'Animation d'Acteurs Virtuels*, PhD thesis, Swiss Federal Institute of Technology at Lausanne (EPFL), Lausanne Switzerland, 1998.

Boulic R., Rezzonico S., Thalmann D., *Multi Finger Manipulation of Virtual Objects*, Proceedings of the ACM Symposium on Virtual Reality Software And Technology, VRST, 67-74, 1996.

Caicedo A., Thalmann D., *Intelligent Decision making for Virtual Humanoids*, Workshop of Artificial Life Integration in Virtual Environments, 5th European Conference on Artificial Life, pp.13-17, Lausanne, Switzerland, September 1999.

Emering L., *Human Action Modeling and Recognition for Virtual Environments*, PhD Thesis, Swiss Federal Institute of Technology-EPFL, Lausanne, Switzerland, 1999.

Funge J., *AI for Games and Animation: A Cognitive Modeling Aproach*, A.Peters, Natick, MA, 1999.

Haton J-P et all, *Le raisonnement en Intelligence Artificielle: techniques, modèles et architectures pour les systèmes à base de connaissances*, InterEditions, Paris 1991.

Macedonia M., Brutzman D., Zyda M., Pratt D., Barham P., Falby J., Locke J., *NPSNET: A Multi-Player 3D Virtual Environment Over The Internet*, Proc. 1995 Symposium on Interactive 3D Graphics, NY:ACM, pp. 93-94, 1995.

Tolani D., Badler N., *Real-Time Inverse Kinematics of the Human Arm*, Presence 5(4), 393-401, 1996.

Turban E., *Expert Systems and Applied Artificial Intelligence*, Macmillan Publishing Company, 1992.

Wiley D., Hahn J., *Interpolation Synthesis for Articulated Figure Motion*, Virtual Reality Annual International Symposium, Albuquerque, New Mexico, March, 1997.