Hierarchical Model for Real Time Simulation of Virtual Human Crowds

Soraia Raupp Musse and Daniel Thalmann
{soraia,thalmann}@lig.di.epfl.ch
Computer Graphics Lab. Swiss Federal Institute of Technology
EPFL, DI-LIG, CH 1015 Lausanne, Switzerland
http://ligwww.epfl.ch

Abstract - This paper describes a model for simulating crowds of humans in real time. We deal with a hierarchy composed of virtual crowds, groups and individuals. The groups are the most complex structure that can be controlled in different degrees of autonomy. This autonomy refers to the extent to which the virtual agents are independent of the user intervention and also the amount of information needed to simulate crowds. Thus, depending on the complexity of the simulation, simple behaviors can be sufficient to simulate crowds. Otherwise, more complicated behavioral rules can be necessary, and in this case, it can be included in the simulation data in order to improve the realism of the animation. We present three different ways for controlling crowd behaviors: i) by using innate and scripted behaviors; ii) by defining behavioral rules, using events and reactions, and, iii) by providing an external control to guide crowd behaviors in real time. The two main contributions of our approach are: the possibility of increasing the complexity of group/agent behaviors according to the problem to be simulated, and the hierarchical structure based on groups to compose a crowd.

Index Terms - crowd motion, degrees of autonomy, reactive behaviors, behavioral animation, synthetic autonomous agents, virtual human crowds.

1 INTRODUCTION

Individual behavior in its most elementary form can be difficult to model. As soon as we consider several individuals and the inter-relationships between them, the complexity of the system is substantially increased. In addition, the mass behaviors and motion of a large number of people have been studied and modeled in computer with different purposes. Some examples of crowd applications are the simulation of a realistic motion of numerous virtual people, which can be directed, as easily as only one agent and the emergent crowd motion simulation providing the evacuation of people in complex environments.

Yet, if the application is to be processed in real time, e.g. integrated in a CVE (Collaborative Virtual Environment) including interactive aspects, a reasonable frame rate is required. This paper presents ViCrowd, a behavioral-based multi-level framework to simulate virtual human crowds in real time. At the end of this paper, the time processing for the different levels of control that are discussed in this paper, are presented.

First of all, we present some useful concepts that have been assumed in this work. A *virtual human agent* (here after referred to as an agent) is a humanoid whose behaviors are inspired by those of humans' [16]. The term *group* is used to refer to a group of agents whereas *crowd* is defined as set of *groups*. Crowd, groups and agents constitute the *entities* of the simulation. *Intentions* corresponds to goals of the entities (e.g. go to the bank), *beliefs* represents internal status of the entities (e.g. emotional status) while *knowledge* describes information of the virtual environment where the crowd is to be simulated (e.g. location of obstacles). *Crowd behavior* corresponds to a set of actions applied according to entities' intentions, beliefs, knowledge and perception. Finally, *events* represent the incidence of something causing a specific *reaction*, which can change the group behaviors.

The problems we deal with in this paper include:

1. Modeling of crowd information and hierarchical structure, also concerning its distribution among groups;
2. Different levels of realism, in order to provide simple crowd behaviors, as well as complex ones;
3. The required structure to provide interaction with groups of agents during the simulation in real-time.

The proposed solution addresses two main issues: i) crowd structure and ii) crowd behavior. Considering ***crowd structure***, our approach deals with a hierarchy composed of crowd, groups and agents, where the groups are the most complex structure containing the information to be distributed among the individuals. Concerning ***crowd behavior***, our virtual agents are endowed with different levels of autonomy. They can either act according to an innate and scripted crowd behavior (*programmed behavior*), react as a function of triggered events (*reactive or autonomous behavior*) or be guided by an interactive process during simulation (*guided behavior*). We introduced the term <guided crowds> to define the groups of virtual agents that can be externally controlled in real time [18].

The next section gives a brief overview of previous work. Section 3 discusses our model to create virtual crowds. In section 4, we introduce the *programmed crowd* while section 5 describes the concept of *reactive behavior*. Section 6 presents how to guide and interact with the *guided crowds*. The discussion closes with some results obtained, as well as with possible future lines of work.

2 RELATED WORK

Previous research on behavioral modeling has mainly focused on various aspects of human control, such as particle systems, flocking systems and behavioral systems. Table 1 has been modified from the original version [20] to include our approach (ViCrowd Model).

METHOD	PARTICLE SYSTEMS	FLOCKING SYSTEMS	VICROWD MODEL	BEHAVIORAL SYSTEMS
Structure	Non-hierarchical	Levels: flock, agents	Levels: crowd, groups, agents	Can present hierarchy
Participants	many	some	many	few
Intelligence	none	some	some(*)	high
Physics-based	yes	some	no	no
Collision	Detect& respond	avoidance	avoidance	avoidance
Control	Force fields, global tendency	Local tendency	Different degrees: pre-defined behavior, rules and guided control	rules

Table 1: **Table modified from the original [20] in order to present various approaches in the area. (* The intelligence in ViCrowd Model can vary from "none" to "some" depending on the rules specified in the conditional events).**

In recent works, Bouvier [6], Brogan and Hodgins [7][8] have used *particle systems* and *dynamics* respectively for modeling the motion of groups with significant physics. Reynolds [22][23] described a distributed behavior model for simulating *flocks* formed by actors endowed with perception skills and the Helbing's team [26] describes methods to simulate the movement of pedestrians. Mataric [15] and Noser [19] developed local rules for controlling collective behaviors. Tu and Terzopoulos [25] have worked on *behavioral animation* for creating artificial life, where virtual agents are endowed with synthetic vision and perception of the environment. Blumberg [3] presented the problem of building autonomous animated creatures for interactive virtual environments, which are also capable of being directed at multiple levels, and Perlin [21] describes IMPROV a system for scripting interactive actors.

Recently, PDI [27] and PIXAR [28] have presented some efforts to simulate crowds of ants in an automatic way. For producing AntZ, two systems were built in order to provide tools to be used for controlling the background characters (over 60.000) with almost the same flexibility as they had with main characters. The Crowd Simulator System is used to produce motion for up to thousands of characters, taking into consideration a combination of physical forces (flow fluids, obstacles, goals, etc) and procedural rules (flocking behaviors and finite-state machine). The Blending System aimed at simplifying the designers' work to avoid creating and assigning individual actions to one character in a crowd. [27].

For the 'Bugs Life' production [28], 4466 different individual motions were created (named "alibis") to describe 228 different behaviors, like nervous, curious, laughing, cheering, clapping, running, walking and panicking. The Pixar team developed a system that allowed alibis to be pieced together in a fluid and flexible way.

2.1 The Context of ViCrowd

ViCrowd aims at presenting a model to automatically generate human crowds based on groups, instead of individuals. Comparing with other works in the domain, "autonomy" and "intelligence" are always present in the individuals, while ViCrowd represents a group-based model. In our case, the groups are more "intelligent" structures, while individuals follow the groups specification. This decision is due to real time requirements of our work and aims to optimize the information needed to provide intelligent agents.

ViCrowd is based on *Flocking systems*, but also includes a simple definition of behavioral rules using conditional events and reactions (see Table 1). Also, a sociological model [17] was described in order to handle affinities and repulsion effects that can emerge in crowd simulation. Some description of this model is presented in section 3.2.2.

Thus the user can decide to use a sociological model and conditional events in order to create more complex behaviors, as well as to randomly distribute action and motion to create static and simple behaviors.

The control in our approach is presented in different degrees of autonomy (guided, programmed and autonomous crowds) ranging from totally interactive to totally autonomous control (without user intervention during the simulation). Table 2 presents some characteristics of crowd control types:

BEHAVIOUR CONTROL	GUIDED CROWDS	PROGRAMMED CROWDS	AUTONOMOUS CROWDS
Level of Autonomy	Low	Medium	High
Level of Intelligence	Low	Medium	High
Execution Frame-rate	High	Medium	Low
Complexity of Behaviors	Low	Variable	High
Level of Interaction	High	Variable	Variable

Table 2: **Characteristics of different types of crowd control.**

The guided crowd is directed using an external control meaning the way to interact with agents (e.g. cyber glove, mouse, etc). In our case, ViCrowd is part of a Client/Server system that is called City-Server [24] dedicated to handle crowds. Some different clients of this framework can be implemented independently of ViCrowd, e.g. CYBER-GLOVE (here the user interacts with agents using a cyberglove), TEXTUAL-INTERFACE clients (where the user interacts with agents writing textual commands).

Concerning the reactive nature: conditional events and reactions are included to model behavior rules and obtain different responses from the crowd. Previous works have discussed the

importance of such a reactive nature for providing more realistic behavior [1][2][10][13].

In comparison with others mentioned methods, ViCrowd presents the following contributions:

1. Multi-level hierarchy formed by crowd, groups and agents
2. Various degrees of autonomy mixing very well known methodologies: scripted and interactive control with rule-based behaviors (reactive behaviors).
3. Group-based behaviors, where agents are simple structures and the group are more complex structures.

3 CROWD MODEL

3.1 Crowd Structure

We defined a crowd as a set of groups composed of virtual agents. Our model distributes the crowd behaviors to the groups (GB) and then to the individuals. Further details about this distribution are presented in section 3.3.

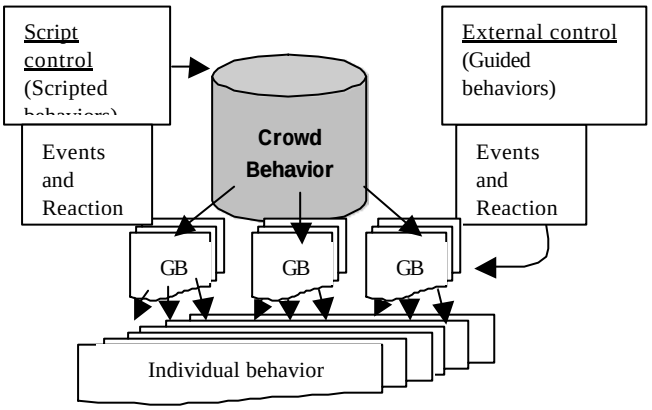

Figure 1: **Hierarchical structure of the model.**

As shown in Figure 1, there are two ways of setting the parameters of our model: scripted and external control. Scripted control defines *scripted behaviors* of the crowd whereas external control specifies *guided behaviors*.

As mentioned before, our crowd model is represented through a hierarchical architecture where the minor entity to be treated consists of groups. In our case, the intelligence, memory, intention and perception are focalized in the group structure. Also, each group can obtain one leader. This leader can be chosen randomly by ViCrowd, defined by the user or can emerge from the sociological rules.

Concerning the crowd control features, ViCrowd aims at providing autonomous, guided and programmed crowds (Table 2). Varying degrees of autonomy can be applied depending on the complexity of the problem. Externally controlled groups, <***guided groups***>, no longer obey their scripted behavior, but act according to the external specification [18].

At a lower level, the individuals have a repertoire of basic behaviors that we call *innate behaviors*. An innate behavior is defined as an "inborn" way to behave. Examples of individual innate behaviors are goal seeking behavior, the ability to follow scripted or guided events/reactions, the way trajectories are processed and collision avoided.

While the innate behaviors are included in the model, the specification of scripted behaviors is done by means of a script language (see Section 5). The groups of virtual agents whom we call <***programmed groups***> apply the scripted behaviors and do not need user intervention during simulation. Using the script language, the user can directly specify the crowd or group behaviors. In the first case, the system automatically distributes the crowd behaviors among the existing groups.

Events and reactions have been used to represent behavioral rules. This reactive character of the simulation can be programmed in the script language (scripted control) or directly given by an external controller (Figure 1). We call the groups of virtual agents who apply the behavioral rules <***autonomous groups***>. Considering the levels of autonomy presented in this work, Figure 2 shows the priority criteria.

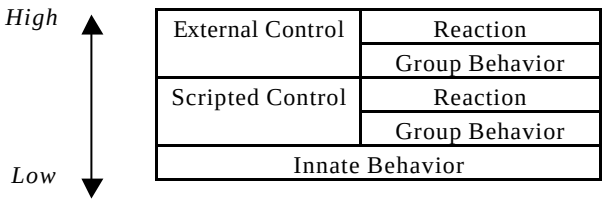

Figure 2: **The behavior priority.**

These priority criteria aim at solving the problems that occur when different types of control are applied at the same time to same characters implying the same nature of tasks. For instance, when the external controller sends an order to go to the restaurant, it can not turn off the collision avoidance, or change the way the trajectories are computed (innate behavior). On the other hand, external controller could explicitly change the group behavior "collision avoidance" to be applied to turn ON or OFF interactively with more priority than the innate behavior.

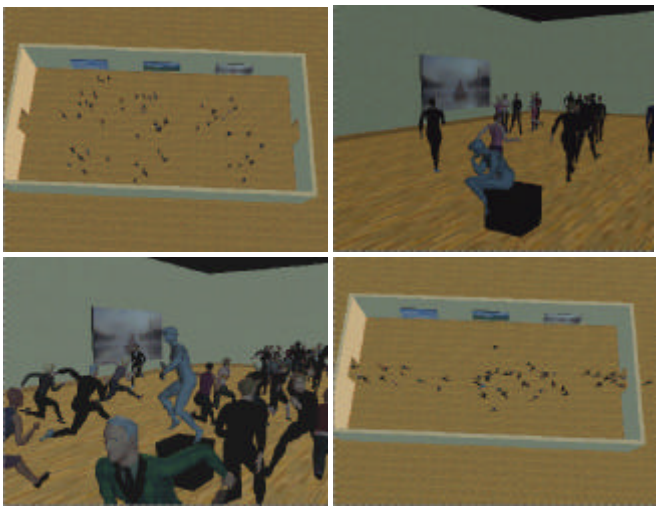

Figure 3: **Scenes of simulation of evacuation due to a panic situation. Up-left and up right: before the panic situation, the crowd walks. Down-left and down right: crowd reacts because an event generated when the statue becomes alive.**

Another example of multiple controls is: if a group's intention is to visit a museum (scripted group behavior), but a panic situation

occurs (event), this group can then perform the programmed reaction associated with the event. This reaction can either be externally specified (during the simulation) or pre-programmed in the script, e.g. exit the environment. Figure 3 shows some images of a panic situation simulation, where 100 agents react by exiting the museum because a statue has become alive. Further details about the reactive behaviors are given in Section 5.

3.2 Crowd Information

We deal with three categories of information in order to characterize the crowds: knowledge, beliefs and intentions. Knowledge represents the information of the virtual environment, for example: <the real position of a chair>. Beliefs describe the internal status of groups and individuals, for instance: <group 0 is happy>. Finally, intention represents the goals of the crowd and groups of agents, e.g. <group 1 goes to the bank>. Table 3 describes the information existent in each one of three levels of entity in our model: (crowd, groups and individuals).

CROWD		**GROUPS**			**INDIVIDUALS**	
Knowledge	*Beliefs*	*Knowledge*	*Beliefs*	*Intentions*	*Beliefs*	*Intentions*
Obstacles to avoid	Crowd parameters	Group memory	Group parameters	Goals and actions to be applied	Relationship with other groups	Follow group or change of groups
Interest points of the scene		Group Perception			Status of domination	Be leader
Actions points						

Table 3: **Categories of information distributed among the entities of crowd and dynamically changed during the simulation**

The next sections present further details about crowd information.

3.2.1 Knowledge

The crowd knowledge represents the information about the virtual environment. Examples of crowd knowledge are locations of the interest points of the scene and information about the action to be applied in some locations. Group knowledge concerns the memory of groups related to the past experiences as well as perception related to agents and groups.

3.2.1.1 Crowd Obstacles

The obstacles to be avoided by the crowd are defined in two ways. The first one relies on the declaration of all objects of the scene; the second one concerns the declaration of the areas where the crowd can walk. The information can also be mixed, declaring some regions where the crowd can walk with some obstacles to be avoided.

3.2.1.2 Crowd Motion and Action

In addition to avoiding obstacles, it is possible to define crowd motion and action. Crowd motion is described using goals that can be: interest points (IP – locations where the crowd must pass through) and action points (AP - locations where the crowd can if necessary go and on arrival must perform an action). These points thus define the crowd paths [18]. Basically, the path followed by the crowd is specified using a set of IPs and APs that are associated with the groups of agents. Figures 4 and 5 show IPs and APs respectively.

As the agents from the same group share the same list of AP/IP (group's goals), each time one group arrives in a goal, we computed one different Bézier curve for each individual between the current goal and next one. Then, these curves are stored for each individual only until the end of their application (during the simulation).

The paths for the different agents from the same group can be similar but are never the same because they cannot occupy the same sub-region, as showed in Figure 4.

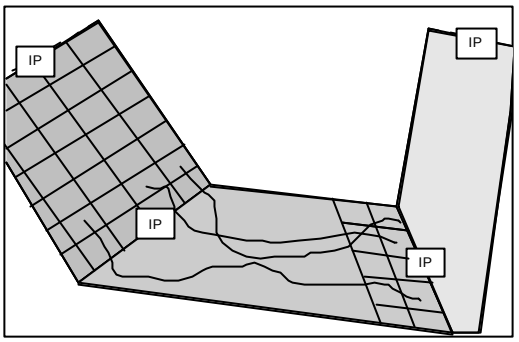

Figure 4: **Family of Bézier curves to define the group paths.**

Figure 5: **Some IPs used to drive the crowd motion.**

Figure 6: **An AP where the action is applause.**

The APs parameters are checked to know if the action to be applied can be used by more than one agent at the same time or not. For instance, a counter (which means in our context, a flat surface in a bank or shop where individuals can be served) is considered as an individual AP whereas a piece of art in a Museum is considered as a shared AP. This classification is useful in order to coherently distribute the agents in AP regions.

3.2.1.3 Group Knowledge

The memory of groups is processed only by the leader of the group. In fact the memory is a structure where the leader's perceived information can be stored and processed afterwards depending on the specified behavioral rules. The size of the memory (capacity of storage) can be pre-defined for each group of crowd.

Group perception concerns the information about the location of groups/agents as well as some associated parameters: knowledge, beliefs and intentions. In this way, a group can perceive the internal status of another group, for instance. As with memory, the perception is associated to just the leader of group.

3.2.2 Beliefs

The crowd beliefs represent the list of behaviors to be applied by the groups as well as the emotion. The crowd beliefs are used to generate the group beliefs. These can be shared specifications or be redefined. The individual beliefs concern internal variables used by the sociological model in order to specify crowd effects: For instance, <relationship with others groups> describes a value for relationships with all the groups of the crowd which can influence the agents' intention to change groups or not (more details about agents' ability to change groups in **Goal Changing** behavior). The parameter <status of domination> represents the individual intention to be a leader or not. These two parameters are only used if the simulation has to apply sociological effects.

3.2.2.1 Crowd and Group Behaviors

High-level behaviors are specified in order to characterize crowds. These behaviors can be programmed in the script language or directly informed using guided control (Fig. 1). The list above presents the eight group behaviors actually existent in ViCrowd.

---

1. Flocking: Group ability to walk together in a structured group movement where agents from the same group walk at the same speed towards the same goals [18]. This behavior is responsible for flocking formation presented in some group motions in the real world, e.g. flock of birds. In our case, we defined four rules to model the flocking formation.

1. the agents from the same group share the same list of goals;
2. they walk at similar speeds;
3. they follow the paths generated as showed in section 3.2.1;
4. one agent can wait for another on arrival at a goal when another agent from the same group is missing.

Consequently, the agents from the same group walk together. We considered it as an important characteristic of our model, because in real life people walk in groups. To decide whether one agent must wait or not for another (rule 4), it is necessary to evaluate if all the agents from the same group arrived on a specific goal. If not, the agents who have arrived must wait.

---

2. Following: Group ability to follow a group or an individual motion. In this case we have defined the assumption of group goals which can be permanent or temporary. Let be *Group A*, a group which follows *Group B*. If the following motion is permanent, *Group A* adopts the goals information of *Group B* until the end of simulation. If this behavior is temporary, *Group A* shares the list of goals of *Group B* at some periods of the simulation but in a randomly defined manner.

---

3. Goal Changing: Agents can have the intention to change groups, consequently assuming the goals of its new group. It can occur only when the sociological effects are applied [17]. Basically, individuals have a more complex structure of parameters including: i) a value for the relationship with all groups (value between 0 and 1) and ii) a value for its domination status, which describes how much the considered agent is able to dominate the others (leadership ability). If the relationship with other groups is better than the current group, individuals can change groups. Also, if the individual presents a high value for leadership ability, he/she can become the new leader of the group, which can change the group behavior too.

---

4. Attraction: Groups of agents are attracted around an attraction point. Using a graphical interface, the user draws bi-dimensional regions or selects specific positions where the crowd should be positioned at a specific time. In association with this command, a <look_at> behavior can be added in order to determine the required orientation for each agent. Figure 7 shows the Open Inventor interface through which the regions are specified where the crowd should be located, as well as the attraction point defined by the <look_at> command.

Fig. 7: **Blue boxes representing the regions where the crowd should be located in and the attraction point is the place where the crowd must look at.**

Fig. 8: **The crowd positioned inside the regions looking at the attraction point.**

To distribute the crowd over bi-dimensional regions, we apply a simple spatial distribution method to define the sub-regions where the agents are placed (Fig. 8). In addition, this distribution considers the required crowd density to be simulated (high or low density) depending on the number of agents. Afterwards, the agents are able to walk to their goal avoiding collision with the others.

5. Repulsion: Group ability to be repulsed from a specific location or region. The opposite of attraction behavior, the repulsion behavior relies on the generation of crowd goals outside the area to be expulsed. Consequently, the agents stay outside the area to be avoided and take their next intentions (goals) representing avoid the repulsed area. At the end of repulsion behavior, the agents are again free to walk in all defined areas. Fig. 9 shows two images of a repulsion simulation.

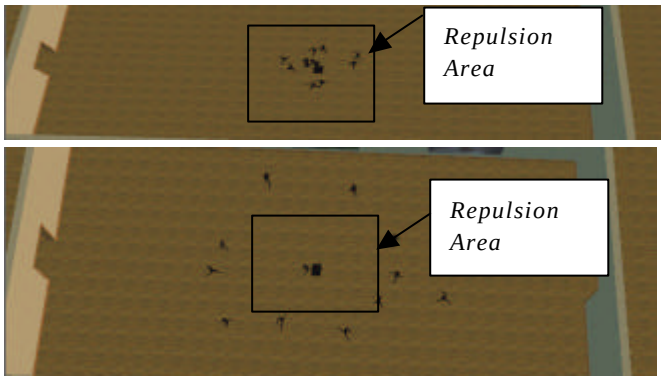

Fig. 9: **Agents are repulsed from a specific location**

6. Split: This behavior concerns the subdivision of a group to generate one or more groups (Figs. 10 and 11). This behavior concerns the randomly generation of intentions to create new groups. The number of agents to be transferred to the new group is random as well as the list of agents. The opposite idea of this behavior (addition of one or more groups) can be programmed using **following behavior.**

Fig. 10: **A group formed by 12 agents has the same intention and walk in the same direction.**

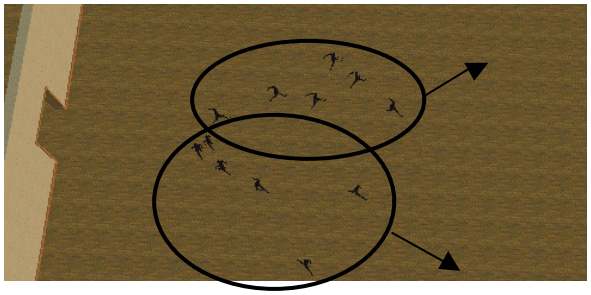

Fig. 11: **The group of Fig. 10 split into two different groups with different intentions.**

7. Space Adaptability: Group ability to occupy all the walking space. The computed trajectory between IPs and/or APs is represented by a Bézier curve. The information considered computing the curve is the region where the IP/AP's are located and divided in sub-regions using a simple spatial distribution method as a function of the required density of people. Adaptability behavior considers the full region to distribute the trajectories. If the agents do not adapt themselves to occupy all the space (without adaptability behavior), the region distributed is smaller and localized close to the goals' location. For example in Figure 12, there are two simulations; the first one represents a group with adaptability behavior and the second one, without it.

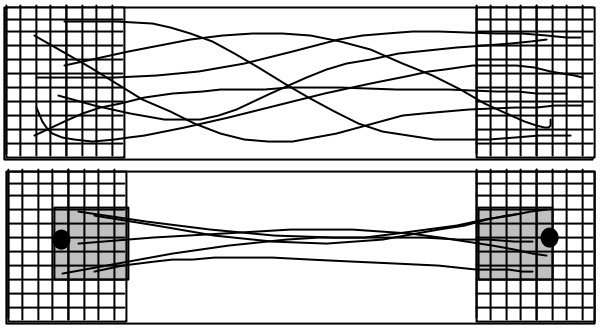

Fig. 12: **(up) All the space information is used to distribute the Bézier curves. 11) (down) A smaller part around the goals' location (black circles) is used in order to determine the trajectories.**

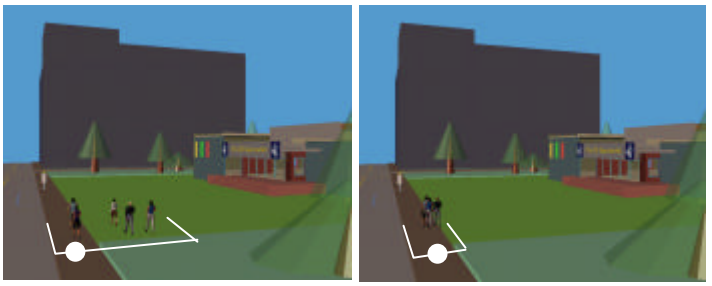

Fig. 13: **(left) Agents walking in all associated region. (right) agents walking in sub-region close to the goal (white circles)**

8. Safe-Wandering: We have used a procedural method in order to evaluate and avoid collision contacts with agents and objects. Our approach is based on the directions changes. The agents can predict the collision event (knowing the position of next virtual humans or obstacles) through simple geometric computing (intersection of two lines). It can therefore avoid the collision by changing its directions through its angular velocity changes.

After a specific period of time, the virtual human returns to its last angular velocity (which was stored in the data structures), and returns to its previous direction.

Fig. 14 shows an image of the collision detection method where agents avoiding collision with other agents and obstacles

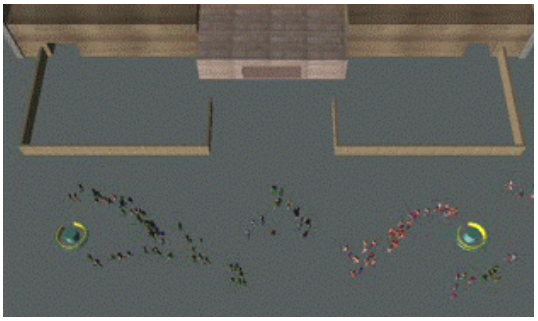

Fig. 14: **Groups avoiding collision with obstacles.**

3.2.2.2 Emotional Status

The emotion property represents the subjective climate to be simulated, e.g., "sad", "calm", "regular", "happy", or "explosive" (here "explosive" means an emotional status happier than happy). These are pre-defined emotional parameters, which can be changed by the user in order to work with other list of names. Depending on this overall definition, the groups are created according to the following parameters: *way of walking, walking speed and range of basic actions.* The correspondence between the names recently cited (sad, calm, etc) and the parameters (e.g., way of walking) is made through a normalized value [0;1].

Using the ViCrowd script language (see Section 4), one can specify how the emotions should be distributed among groups, as shown in Listing 1. When events and reactions are specified, the emotional status can also be used to define a condition (section 5) to trigger an event.

```
CROWD_EMOTION
 PARTY          PERC 80 EXPLOSIVE
```

Listing 1: **Script language to specify crowd emotion.**

Listing 1 shows a crowd of which 80 percent of the people have the emotional status <explosive>. The other 20 percent will be generated in a random way. Yet, the emotional status for specific groups can also be redefined. For example, it is possible to simulate a <happy> crowd with one or several <sad> groups. In the same way, the emotion can be changed as a function of triggered events and reactions. Figures 15 and 16 show some postures and ways of walking [4][5] taken on by the crowd according to their emotion.

Fig. 15: **Different postures as a consequence of various emotions.**

Fig. 16: **Different ways of walking as a result of various emotions.**

3.2.2.3 Individual Beliefs

As mentioned before, we consider the individual agents more simple than the groups of agents. While the groups contain goals, emotions, beliefs, knowledge and intentions, the individuals are just able to walk whereas avoid collision with obstacles and other agents. However, when the sociological effects are applied, it's possible for the individuals to have a more complex structure of parameters including: i) *goal-changing behavior,* group behavior (see Section 3.2.2.1); ii) *a value for the relationship with all groups* (value between 0 and 1) and iii) *a value for its domination status* describing how much the considered agent is able to dominate the others (leadership ability).

3.2.3 Intentions

The crowd does not have intentions (see Table 3), unless the group intentions are not specified. In this case, the crowd knowledge is used to generate crowd intentions, which afterwards are used to generate group intentions in a random way. For example, the crowd knowledge can define some interest points (IPs) and action points (APs). If the group intention is not defined, ViCrowd computes for each group a list of IPs and APs to be followed. The individual intention determines if the agent will follow its group's specification or change groups, for example exiting $Group_i$ and joining $Group_j$. The other individual intention is dependent of the <domination value> (individual belief) specified for each agent. If this value (between 0 and 1) is the highest of its group, this agent can became the new leader.

3.2.4 Inter-Relationship between the Various Categories of Information

The three types of information distributed in the multi-layered architecture can present some inter-dependence. The *knowledge* represents the information coming from the virtual environment and this can be used together with *beliefs* in order to apply different *intentions.* For example if a group has the *intention* to <go to the bank> and the agents from this group have the *belief* to <follow the group>, they are able to go to the bank if the crowd *knowledge* contains the information about <where is the bank>. Figure 17 shows the general graph of internal dependence between the various levels of information. The knowledge cannot be changed except if it is made through an external control during the simulation. The beliefs and intentions are dependent on one another, and also dependent on knowledge.

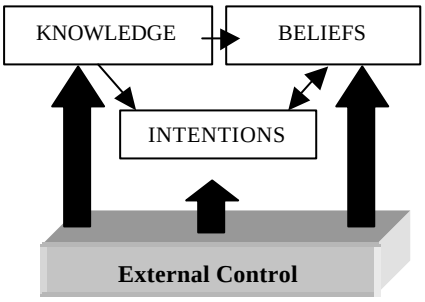

Fig. 17: **Inter-dependence between the levels of information**

3.3 Overview of ViCrowd Architecture

This section presents the overview of ViCrowd architecture. The crowd information together with perceptions, innate and sociological behaviors and treatment of events are processed by "Behavioral Motor" (BM) in order to achieve the groups and individuals low-level behaviors. BM is responsible by performing the following function:

```
Low_level_behavior = (   perception, innate_ability,
                         group_information,
                         sociological_behavior,
                         events_treatment,
                         priority_rules          )
```

Figure 18 presents the ViCrowd architecture.

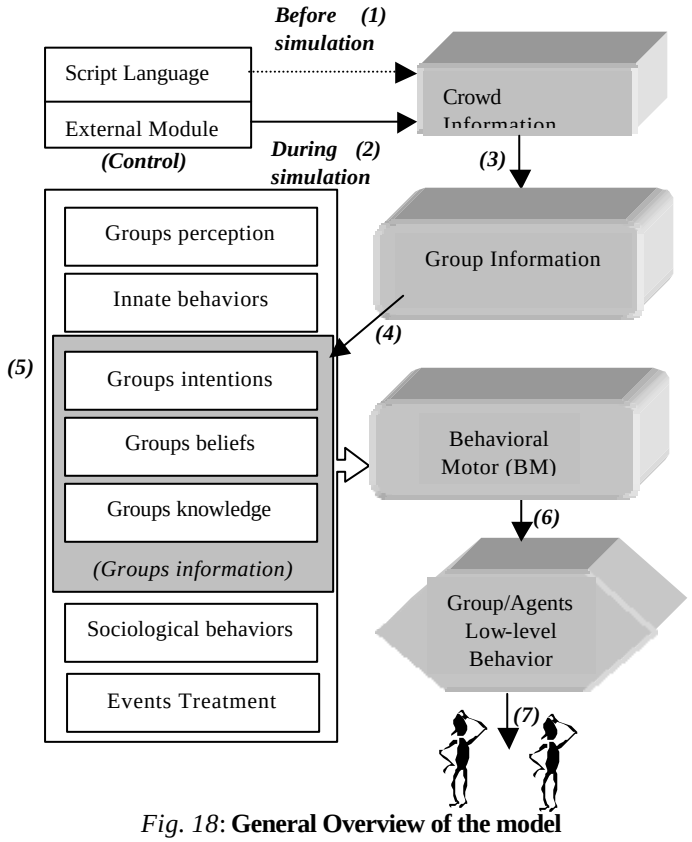

Fig. 18: **General Overview of the model**

In (1) the scripted behaviors are defined before the simulation, (2) describes the process from which one is able to interact and to send orders to the virtual crowd. In (3) the crowd information described in (1) and (2) is distributed among groups. The generation of group information can take into account three different aspects:

- Information specified based on the script language
- Information specified based on the external controller
- Information emerged from the behavioral rules

In (4) the group information describes the knowledge, intentions and beliefs associated with other information like events and sociological behaviors (5). In (6) the BM is represented concerning the process where the group and individual low-level behaviors are generated, e.g. current goal, current speed, current emotional status, etc.

The low-level behaviors generated for groups/individuals are (7):
- Internal status of individuals and groups which deals with the way of walking, walking speed and repertory of basic actions
- Goals (motion, action).

The decision behind this process is very simple and involves the application of priority rules following the following steps:

1. Analyzing of group information (informed in the script language or by the external control), the ***low-level behaviors are generated*** for each group.
2. If sociological effects are applied***, low-level behaviors of groups can be changed*** because agents can change groups and agents can become new leaders of the groups.
3. If triggered events occur, the reaction applied (informed in the script language or by the external control) can change **low-level behaviors of the groups** too.
4. Finally, the groups' low-level behaviors are distributed among the individuals, which use perception and their innate abilities to achieve the groups' intentions. All the individuals in a group have similar parameters and always follow the group tendency.

4 PROGRAMMED BEHAVIORS *(PROGRAMMED CROWD)*

A script language has been used in order to program specific behaviors in ViCrowd. There are three kinds of information to be dealt with in the script language:

1. Geometrical information, like IPs and APs specification
2. Behavioral information, like groups' behaviors specification and groups' intentions
3. Simulation information that deals with parameters of simulation. For instance, type of interpolation used in the trajectories [18], the generation of a log file of simulation, etc.

ViCrowd script deals with pre-defined crowd-based commands. So, it is not oriented to provide a programming language like C++, for instance. The main goal of ViCrowd script is to provide pre-defined commands to easily specify crowd behaviors. Also, keeping defaults values in all commands achieve the possibility of programming crowds with few parameters.

Considering the geometrical information, IPs and APs can be declared in order to define the motion and the action of crowd, respectively. This is the basic information ViCrowd requires in

order to deal with the groups motion. Listing 2 shows an example of IP and AP generated using the graphical interface.

```
IP0       POSITION ( 1000 0 2500 )
ORIENTATION ( 0.87 0 –0.4 )
ASSOCIATED_REGION 3 PTS:  ( 1300 0 5600 )  ( 2100 0 4200 )
                                ( 1250 0 200 )
AP0       POSITION ( 1000 0 2500 )
ORIENTATION ( 0.87 0 –0.4 )
ASSOCIATED_REGION 2 PTS:  ( 1300 0 5600 )  ( 2100 0 4200 )
TYPE SHARED
ACTION PARAMETERS KEYFRAME applause
```

Listing 2: **ViCrowd script to generate IPs and APs.**

IP0 is the name of an interest point (IP) which is placed in POSITION. The agents should apply ORIENTATION when they arrive in this IP in addition to be inside the ASSOCIATED_REGION, formed by three points. AP0 geometrical information contains the same kind of information as IP0, added to action parameters. TYPE SHARED represents an AP which agents can apply programmed actions at the same time (for example, applause). The pre-defined action sets the keyframe motion to be applied (e.g. applause).

The behavioral information concerns information of the groups' behaviors. This information can be explicit to the groups or generic to the crowds. In the last case, ViCrowd distributes this information among groups. *Listing 3* shows an example of crowd behavior parameters.

```
CROWD_SEMANTIC
INTENTION
  PERCENTAGE 20 GOES TO IP0 APPLIES AP0
  PERCENTAGE 40 DOES NOTHING
BELIEFS
        EMOTION ALL HAPPY
        LEADER PERCENTAGE 10
GROUP_BEHAVIORS
         ADAPTABILITY PERCENTAGE 20
         FLOCKING GROUP_5
SET SOCIOLOGICAL_EFFECTS ON
```

Listing 3: **ViCrowd script to define crowd behaviors.**

Considering that knowledge was already defined using IPs and APs (See *Listing2*), the crowd intentions and beliefs can be specified as shown in Listing 3. In this case, we can define the behavior of the whole crowd (command ALL), the percentage of people who have to apply some behaviors (command PERCENTAGE) or the specific group who should apply some behavior (e.g. FLOCKING in Listing 3). For information that is not specified, ViCrowd handles this with random behaviors.

5 REACTIVE BEHAVIORS *(AUTONOMOUS CROWD)*

We define the reactive nature of a crowd as the ability to react to events. An event is a structure that can trigger a reaction. The contribution of this method is the possibility of creating various kinds of events such as: *general events*, affecting all agents e.g., a panic situation; *local events*, affecting agents in a determined location e.g., agents at a bus stop; and *emotional events*, affecting agents depending on their emotional status. An example of an emotional event simulation is given in section 5.3.

One or several events can trigger a reaction that can in turn activate one or several events. Thus, it is possible to simulate a chain reaction, as in the dialog simulation presented in section 7.1. An event can be randomly generated by the system, explicitly specified in the script language, activated as part of a programmed reaction or extemally activated in real time.

5.1 Event

The information included in the event structure consists of <*when*> the event must occur and <*who*> it should affect. Listing 4 shows an example of a conditional event generated when any group is located at the bus stop. The bus stop is modeled as an interest point (IP) called IP_BUS (similarly presented in Listing 2).

```
event BUS_STOP
WHEN    CONDITION   (Object_bus NEAR  IP_BUS)
WHO      ALL_GROUPS NEAR IP_BUS
```

Listing 4: **An example of a script language defining an event called <Bus_stop>.**

5.2 Reaction

The reaction concept contains the definition of the behaviors to be applied if the associated event occurs. Some parameters that can describe reactions assumed by the affected groups are <motion>, <action>, <posture>, <internal_status_changing>, etc. The reaction <Take_the_bus>, triggered by the event <Bus_stop> (Listing 4), can be defined as follows:

```
Reaction TAKE_THE__BUS
                ACTION KEYFRAME take_bus
                MOTION ATTACH Object_bus
                CHANGE_EMOTION HAPPY
```

Listing 5: **Example of a script language for specifying a reaction.**

The reaction <Take_the_bus> is composed of a keyframe sequence [4][5] to be played, and the motion of the group must be linked to the <Object_bus> motion. After the reaction <TAKE_THE_BUS> is applied, the agent has the emotion equal to HAPPY.

5.3 Case-study using Reactive Behaviors to Simulate Behavioral Rules - Simulating a party

In this example, the objective is to simulate a party, where agents look for food if they are hungry (i.e. agent status). Otherwise, the agents have a status that we shall call "social", meaning walking and meeting others. Initially, the crowd is divided into 70 percent of the population having a <hungry> status, and the rest, a <social> status (see Listing 6).

```
 CROWD_STATUS
 PARTY        PERCENTAGE 70 HUNGRY
              PERCENTAGE 30 SOCIAL
```

Listing 6: **Script language to specify crowd status.**

The behavioral rules are presented in Listing 7 and Listing 8:

The agents whose status is equal to hungry, look for food when the food arrives.

```
event: HUNGRY
        WHO GROUP_STATUS HUNGRY
        WHEN EVENT FOOD MATCHED
reaction HUNGRY
        MOTION GO TO TABLE /* TABLE = IP */
        ACTION INTERACTION SOBJ TABLE
```

Listing 7: **Script language for defining of a simple rule.**

In Listing 7, the command <INTERACTION SOBJ> calls an external module responsible for the management of interaction between agents and smart objects [14].

The agents that have already eaten have their status changed to social.

```
event CHANGE
        WHO GROUP_STATUS HUNGRY AND
        WHEN INTERACTION SOBJ TABLE MATCHED
                AND REACTION HUNGRY MATCHED
reaction CHANGE
        GROUP_STATUS SOCIAL
```

Listing 8: **Script language for defining a simple rule.**

Fig. 19: **The interaction with Smart object [14] occurs for the hungry agents. The other agents continue walking.**

After application of these rules, when more food is brought, only the hungry agents that did not eat previously will respond to the event. Figure 19 shows a scene from this simulation.

6 External Behaviors *(Guided Crowd)*

In order to provide external control to the guided crowd, we define a client/server system [12] to combine a Rule-Based Behavior System (RBBS) [24] and ViCrowd. The RBBS is able to control groups or individuals by sending the information specifying what motion, action, event or reaction is to be applied. The input of the RBBS consists of behavioral rules in a simple syntax similar to the English language (Listing 9).

```
rule2: if ((Laura needs to take the train and (Laura has no ticket))
        then (Laura goes to counter to buy one)
```

Listing 9: **Example of RBBS Rules and Events defined by user.**

In order to translate the RBBS information for ViCrowd clients, the server uses a database (DB World) that stores the environment information (e.g. location of a bus stop, size of a chair, etc). Thus, the server is able to translate and transmit the codified order into a command readable by the crowd client. The following tables show an example of various possible degrees of autonomy during the simulation of group_1. Table 4 concerns scripted behaviors, and table 5 summarizes the guided behaviors to be sent during simulation.

Identifier	Scripted Behavior
S0	Group_1 go to shopping – buy a hotdog

Table 4: **The scripted behavior for group_1.**

Identifier	Simulation time	Guided behaviors
G0	101	Group_1 go to the bank

Table 5: **The guided behavior sent by RBBS.**

Table 6 lists the control changes that occur during a simulation period of group_1. An important feature in this example is the usage of group memory in order to store information about the previous behaviors (achieved or not).

Simulation time	***Group_1 control***	***Group_1 status***	***Scripted behaviors - SB -***	***Guided behaviors - GB -***
0	Scripted	Acting S0	S0	
101	Guided	Save S0 onto the group memory		G0
102	Guided	Acting G0		G0
150		Finished Action G0 – Take the last not completed action (S0)		
151	Scripted	Acting S0	S0	
200	Scripted	Finished Action S0		

Table 6: **The exchanging of the group_1 control**

When a guided behavior (time 101) arises and a scripted action has not yet terminated, SB is stored in the group memory in order to be re-activated later (time 151). A guided behavior has priority over all scripted behaviors.

7 Results

One goal of this research is to demonstrate that simulations of crowds can be done in general virtual environments. We have chosen some examples in order to test our crowd model, including a train station, a virtual city, a park and a theatre.

7.1 Applications

The virtual city is a geometrical and semantic environment (see Fig. 20) [12]. The train station, the park and the theatre are part of this project. To simulate the crowd in the virtual city, we insert IPs and APs in order to program the crowd motion.

Figure 20: **Simulation in a park in the virtual city [12].**

The train station simulation includes many different actions and places, where several people are present and doing different things. Possible actions include “buying a ticket”, “going to shop“, ”meeting someone”, “waiting for someone”, “making a telephone call”, “checking the timetable”, etc. This simulation uses external control (RBBS [12][24]) to guide some crowd behaviors in real time (see Figures 21 and 22).

Fig. 21: **Train station simulation.**

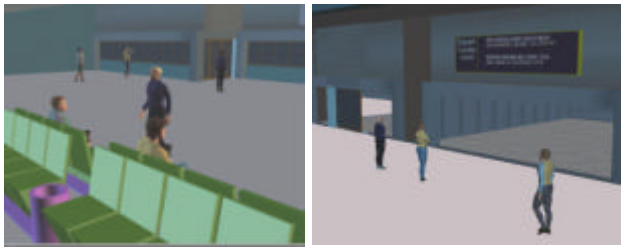

Fig. 22: **Guided behaviors. *On the left:* group going to sit down. *On the right:* group checking the timetable.**

The diverse nature of crowd behavior implies the need for high-level behaviors to simulate a variety of public events. We show two new examples: the first one is a conference attended by a crowd in a theatre. In this case, the user specifies directly in real time events generated by the speaker whereas the reactions are pre-programmed in the script. For example, the user can trigger an event that should generate a positive reaction, the latter being programmed in the script, e.g. “applaud”. Figure 23 shows an image of the simulation of reactive crowds in a theatre.

Fig. 23: **Crowd in a theatre.**

The second example is set in a park in the virtual city [12]. Using reactive behaviors, the dialogue between the leader and the crowd was programmed in order to establish a communication link. Figures 24 and 25 show images of this simulation.

Fig. 24: **Dialogue between the leader and the crowd. On the left, the leader talks and on the right, the crowd reacts.**

Fig. 25: **Crowd reacting.**

7.2 Discussion

We have presented algorithms that allow an animator to simulate crowds in general environments with various levels of behaviors realism in real time. The four examples given in section 7.1 highlight the different ways of controlling the simulations. The crowd control used to populate the virtual city is the scripted behaviors (SB). The train station simulation applies SB, and guided behaviors (GB) when external control occurs (EC). The dialog simulation employs SB and reactive behaviors (RB) when some event occurs, whereas the theatre simulation involves SB, GB and RB. Figure 26 compares these four different controls for simulating crowds.

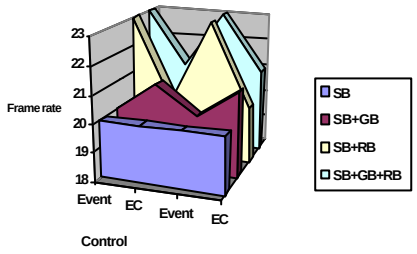

Fig. 26: **Frame rate considering different crowd controls (*Event* represents the time when a RB – *Reactive Behavior* - happens, and *EC - External Control* – represents the time when a *GB – Guided Behavior* - happens)**

The frame rate for simulating a crowd of 100 agents for 1000 frames on an Onyx 2 (SGI) is shown in Table 7.

Control	Execution Frame rate
SB	20
SB+GB (during 500 frames)	19
SB+RB (during 500 frames)	17
SB+GB(during 500 frames) + RB (during 500 frames)	16

Table 7: **Frame rate for processing 100 agents over 1000 frames with different levels of control.**

CONCLUSIONS

We have described in this paper the ViCrowd model which aims to simulate crowds in real time with different levels of autonomy. A novel idea of this paper is to present the possibility of generating simulations with various levels of realism including scripted, reactive and guided behavior, depending on the required compromise for each application. Another contribution of this paper is the hierarchical structure used to model crowds based mainly on groups, instead of individuals. Comparing our group-based approach with an individual approach, we believe that the sense of crowd behavior and presence have been achieved in ViCrowd where families and groups of friends can be observed. We are currently investigating the simulation of crowd behaviors in panic and emergency situations. We are interested in modeling the crowd motion as well as the sociological behavior of the people in these conditions.

For future work, we are interested in improving the facility to use the ViCrowd script language by incorporating it in a programming language like Python, for instance.

Some examples of animation can be seen in http://ligwww.epfl.ch/~soraia.html.

ACKNOWLEDGMENTS

The authors are glad to thank the to TVCG reviewers who have collaborated so much in the revision of this paper, as well as Cher Richardson for English proof-reading. This research was sponsored by the Swiss National Research Foundation, the Federal Office of Education and Science in the framework of the European project eRENA [11], FUNDEPE and CAPES - Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Brazilian offices of Education and Science).

REFERENCES

[1] Arkin, R. A., "Integrating behavioral, perceptual and world knowledge in reactive navigation". In P. Maes (ed.), Designing Autonomous Agents, pp. 105-122, Cambridge, MA: MIT Press, 1990.

[2] Arbib, M. A. & Lee, H. B., "Anural visuomotor coordination for detour behaviour: from retina to motor schemas". In J.-A. Mayer, H. L. Roitblat and S. W. Wilson (ed.), From Animals to Animats II, Cambridge, MA: MIT Press, 1993.

[3] Blumberg, B.; Galyean, T. "Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments". SIGGRAPH - Computer Graphics Proceedings, pp 47-54. Los Angeles, 1995.

[4] Boulic, R; Capin, T.; Huang, Z.; Kalra, P.; Lintermann, B.; Magnenat-Thalmann, N.; Moccozet, L.; Molet, T.; Pandzic, I.; Saar, K.; Schmitt, A.; Shen, J. and Thalmann, D. "The HUMANOID Environment for interactive Animation of Multiple Deformable Human Characters". Proceedings of EUROGRAPHICS'95, p.337-348 (Maastricht, The Netherlands, August 28 september, 1995).

[5] Boulic, R.; Huang, Z.; Thalmann. D. "Goal Oriented Design and Correction of Articulated Figure Motion with the TRACK System". Journal of Computer and Graphics, v.18, n.4, pp. 443-452, Pergamon Press, October 1994.

[6] Bouvier, E.; Cohen E.; and Najman. L. "From crowd simulation to airbag deployment: particle systems, a new paradigm of simulation". Journal of Electronic Imaging 6(1), 94-107 (January 1997).

[7] Brogan, D. and Hodgins, J. "Group Behaviours for Systems with Significant Dynamics". Autonomous Robots, 4, 137-153. 1997.

[8] Brogan, D.C., Metoyer, R.A. and Hodgins, J.K. "Dynamically simulated characters in virtual environments". In IEEE COmputer Graphics and Applications. Vol.18, No5, pp 58-69. Sept. 1998.

[9] DIVE - http://www.sics.se/dive/

[10] Drogou, L.A and Ferber, J. "Multi-agent simulation as a tool for studying emergent processes in societies". Gilbert, N and Doran, J. editors, In Proceedings of Simulating Societies: the computer simulation of social phenomena, North-Holland, 1994.

[11] eRENA - http://www.nada.kth.se/erena/index.html

[12] Farenc, N., Musse, S.R, Schweiss, E., Kallmann, M., Aune, O., Boulic, R and Thalmann, D. "A Paradigm for Controlling Virtual Humans in Urban Environment Simulations". Special Issue on Intelligent Virtual Environments, 1999.

[13] Giroux, S. "Open Reflective Agents". M.; Wooldridge, J.P. Muller and M.; Tambe, editors, Intelligent Agents vol. II, Agent Theories, Architectures, and Languages, pp. 315-330. Springer-verlag, lnai (1037) edition, 1996.

[14] Kallmann, M. and Thalmann, D. “Modeling Objects for Interaction Tasks”. Proc. Eurographics Workshop on Animation and Simulation, 1998

[15] Mataric, M. J. “Learning to Behave Socially, in D. Cliff, P. Husbands, J.-A. Meyer & S.Wilson, eds, From Animals to Animats: International Conference on Simulation of Adaptive Behavior, pp.453-462. 1994.

[16] Meyer, J.A.; Guillot, A. “From SAB90 to SAB94: Four Years of Animat Research”. In: Proceedings of Third International Conference on Simulation of Adaptive Behavior. Brighton, England, 1994.

[17] Musse, S.R. and Thalmann, D. “A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis”. Proc. Workshop of Computer Animation and Simulation of Eurographics’97, Sept, 1997. Budapest, Hungary.

[18] Musse, S.R., Babski, C., Capin, T. and Thalmann, D. “Crowd Modelling in Collaborative Virtual Environments”. ACM VRST /98, Taiwan

[19] Noser, H., Thalmann, D. “The Animation of Autonomous Actors Based on Production Rules”. Proc. Computer Animation ’96, 1996, Geneva, Switzerland.

[20] Parent, R. Notes on Computer Animation: “Computer Animation: Algorithms and Techniques”. http://www.cis.ohiostate.edu/~parent/book/outline.html

[21] Perlin, K.; Goldberg, A. “Improv: A System for Scripting Interactive Actors I Virtual Worlds”. SIGGRAPH – Computer Graphics Proceedings. Pp. 205-216. New Orleans, 1996.

[22] Reynolds, C. “Flocks, Herds and Schools: A Distributed Behavioral Model”. Proc. SIGGRAPH ’87, Computer Graphics, v.21, n.4, July, 1987.

[23] Reynolds, C. “Steering Behaviors for Autonomous Characters”. Game Developers Conference, 1999.

[24] Schweiss, E., Musse, S.R.; Garat, F. "An Architecture to Guide Crowds based on rule-based systems". Autonomous Agents '99, Seattle, Washington, USA, 1999.

[25] Tu, X. and Terzopoulos, D. “Artificial Fishes: Physics, Locomotion, Perception, Behavior”. Proc. SIGGRAPH ’94, Computer Graphics, July 1994.

[26] Treiber, M., Hennecke, A. and Helbing, D. “Microscopic Simulation of Congested Traffic”. In: Traffic and Granular Flow’99: Social Traffic, and Granular Dynamics (Springer, Berlim) 1999.

[27] AntZ. http://www.antz.com

[28] Bugs Life. http://bugslife.com