FROM AUDIENCES TO MOBS: CROWD SIMULATION WITH PSYCHOLOGICAL FACTORS

A DISSERTATION SUBMITTED TO

THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Funda Durupınar

July, 2010

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assoc. Prof. Dr. Uğur Güdükbay (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Varol Akman

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Özgür Ulusoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. A. Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assoc. Prof. Dr. Veysi İşler

Approved for the Institute of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Institute

ABSTRACT

FROM AUDIENCES TO MOBS: CROWD SIMULATION WITH PSYCHOLOGICAL FACTORS

Funda Durupınar
Ph.D. in Computer Engineering
Supervisor: Assoc. Prof. Dr. Uğur Güdükbay
July, 2010

Crowd simulation has a wide range of application areas such as biological and social modeling, military simulations, computer games and movies. Simulating the behavior of animated virtual crowds has been a challenging task for the computer graphics community. As well as the physical and the geometrical aspects, the semantics underlying the motion of real crowds inspire the design and implementation of virtual crowds. Psychology helps us understand the motivations of the individuals constituting a crowd. There has been extensive research on incorporating psychological models into the simulation of autonomous agents. However, in our study, instead of the psychological state of an individual agent as such, we are interested in the overall behavior of the crowd that consists of virtual humans with various psychological states. For this purpose, we incorporate the three basic constituents of affect: *personality*, *emotion* and *mood*. Each of these elements contribute variably to the emergence of different aspects of behavior. We thus examine, by changing the parameters, how groups of people with different characteristics interact with each other, and accordingly, how the global crowd behavior is influenced.

In the social psychology literature, crowds are classified as mobs and audiences. Audiences are passive crowds whereas mobs are active crowds with emotional, irrational and seemingly homogeneous behavior. In this thesis, we examine how audiences turn into mobs and simulate the common properties of mobs to create collective misbehavior. So far, crowd simulation research has focused on panicking crowds among all types of mobs. We extend the state of the art to simulate different types of mobs based on the taxonomy. We demonstrate various scenarios that realize the behavior of distinct mob types.

Our model is built on top of an existing crowd simulation system, HiDAC

(High-Density Autonomous Crowds). HiDAC provides us with the physical and low-level psychological features of crowds. The user normally sets these parameters to model the non-uniformity and diversity of the crowd. In our work, we free the user of the tedious task of low-level parameter tuning, and combine all these behaviors in distinct psychological factors. We present the results of our experiments on whether the incorporation of a personality model into HiDAC was perceived as intended.

Keywords: Crowd simulation, autonomous agents, simulation of affect, crowd taxonomy, mob behavior.

ÖZET

KİTLELERDEN GÜRUHLARA: PSİKOLOJİK FAKTÖRLERLE KALABALIK SİMÜLASYONU

Funda Durupınar
Bilgisayar Mühendisliği, Doktora
Tez Yöneticisi: Doçent Dr. Uğur Güdükbay
Temmuz, 2010

Kalabalık simülasyonu, biyolojik ve sosyal modelleme, askeri simülasyonlar, bilgisayar oyunları ve filmler gibi geniş uygulama alanlarına sahiptir. Canlandırılmış sanal kalabalıkların simülasyonu bilgisayar grafikleri camiası için zorlu bir görevdir. Fiziksel ve geometrik özelliklerinin yanısıra, gerçek kalabalıkların hareketlerinin anlamları, sanal kalabalıkların tasarım ve gerçekleştirilmesinde önemlidir. Psikoloji, bizim kalabalıkları oluşturan bireylerin motivasyonlarını anlamamıza yardımcı olur. Özerk etmenlerin simülasyonuna psikolojik modelleri dahil etmek üzerine yoğun araştırma yapılmıştır. Buna rağmen, biz, çalışmamızda bireysel bir etmenin kendisinden ziyade çeşitli psikolojik özelliklere sahip bireylerden oluşan bir kalabalığın genel davranışıyla ilgilenmekteyiz. Bu amaçla, duygulanımın üç temel bileşenini dahil ettik: *kişilik*, *duygu* ve *mizaç*. Bu etkenlerden her biri farklı davranış sekillerinin ortaya çıkmasına farklı derecelerde katkıda bulunur. Böylece, parametreleri degiştirerek, farklı özelliklere sahip grupların birbirleriyle nasıl etkileştiklerini, ve buna bağlı olarak genel kalabalık davranışının nasıl etkilendiğini inceliyoruz.

Sosyal psikoloji literatüründe kalabalıklar, kitleler ve güruhlar olarak sınıflandırılmıştır. Kitleler pasif kalabalıklar, güruhlar ise, duygusal, mantıksız ve görünürde homojen davranışlarda bulunan aktif kalabalıklardır. Bu tezde kitlelerin güruhlara dönüşümünü ve güruhların kolektif olarak uygun olmayan davranışlarda bulunuşunu inceliyoruz. Mevcut kalabalık simülasyonu araştırmaları, tüm güruh çeşitleri içinde sadece panik davranışı gösteren güruhlara odaklanmıştır. Biz, en son gelişmeleri kalabalıkların sınıflandırılmasına göre değişik çeşit güruhların simülasyonunu yaparak genişletiyoruz. Farklı güruh tiplerinin davranışını gerçekleştiren çeşitli senaryolar gösteriyoruz.

Modelimiz, mevcut bir kalabalık simülasyonu sistemi olan HiDAC (Yüksek Yoğunluklu Özerk Kalabalıklar) üzerine kurulmuştur. HiDAC, bize kalabalıkların fiziksel ve alt duzeydeki psikolojik özelliklerini sağlar. Biz çalışmamızda, kullanıcıyı meşakkatli olan alt düzey parametre ayarlama işinden kurtararak bütün bu davranışları farklı psikolojik faktorlerde birleştiriyoruz. Bir kişilik modelinin HiDAC sistemine dahil edilmesi işleminin niyetlendiğimiz şekilde algılanıp algılanmadığına dair yaptığımız deneylerin sonuçlarını sunuyoruz.

Anahtar sözcükler: Kalabalık simülasyonu, özerk etmenler, duygulanım simülasyonu, kalabalıkların sınıflandırılması, güruh davranışı.

Acknowledgement

I would like to express my deepest thanks and gratitude to my supervisor Assoc. Prof. Dr. Uğur Güdükbay for his invaluable suggestions, support and guidance during this thesis research.

It is an honor for me to thank all the committee members for spending their time and effort to read and review my thesis. I am grateful to Prof. Dr. Varol Akman, Prof. Dr. Enis Çetin, Prof. Dr. Özgür Ulusoy, Assoc. Prof. Dr. Veysi İşler and Prof. Dr. Bülent Özgüç for their insightful comments and suggestions.

My sincere thanks also goes to Prof. Dr. Norman Badler, Dr. Nuria Pelechano and Dr. Jan Allbeck for hosting me as a part of their research group. Their collaboration and guidance provided me with invaluable insight throughout my research.

I would like to thank Ates Akaydın for providing the city model that we used in our simulations. His skills and patience enabled the creation of more realistic animations.

I am grateful to the former and current members of the Computer Engineering Department, including my professors and colleagues, who made this department a lovely and warm place. I owe my warmest thanks to my friends Meltem, Deniz, Sengör, Kamer, Engin, Barla, Cihan, Duygu, Nil, Özlem, Ayşe, Tayfun, Oğuzcan, Alper, Murat, Volkan, and Can, whose valuable friendship turned my years in Ph.D. study into the happiest part of my life.

Above all, I would like to thank my family for their love, support and motivation. Without their unconditional support, this thesis would not have been possible.

And finally, my special thanks are for Özgün, who has been there for me with his love, patience, and encouragement.

To my family ...

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Crowd simulation has a wide range of application areas from computer games to evacuation planning for building security. The topic has drawn the attention of computer graphics and visualization community as well as cognitive science and artificial intelligence researchers. Since a human being is a complex structure, masses of human beings should be even more complicated to study. When humans form groups, interaction becomes an essential part of the overall group behavior. In some cases, individuality gets lost and collective behavior comes on the scene. The semantics underlying the motion of real crowds should be studied extensively in order to achieve realistic behavior in virtual ones. Therefore, crowd simulation research also benefits from social psychology literature.

Our main purpose is to understand the basics of crowd psychology and build our model on scientific grounds. There has been extensive research on incorporating psychological models into the simulation of autonomous agents. Most of the emphasis in this field is put on individual agents, usually conversational, interacting with a human user. However, we are not interested in the behavior of an individual per se but the incorporation of a psychological model into large groups of people. We thus examine, by changing the parameters, how subgroups of people with different psychological traits interact with each other, and accordingly, how the global crowd behavior is influenced.

Sometimes, regular crowds start to act collectively, showing highly emotional and illogical behaviors. Crowd psychology has been widely investigated by social psychologists. Researchers have come up with different theories to explain the collective craze. These theories range from formulating this phenomenon by the loss of individuality through contagion to predisposition hypotheses. Crowd simulation community, on the other hand, has not focused on this aspect of crowds except panic situations and egress scenarios. However, regular crowds can turn into various types of mobs, showing different emotions such as anger or even euphoria. Classification of mobs can also be found in the social psychology literature.

1.2 Contributions

This thesis study contributes to the literature in two parts. The first part is the incorporation of a psychological model into the virtual agents in the crowd.

The components making up the psychological state are personality, emotion and mood. Research so far has focused on incorporating an affect model into conversational or interactive virtual agents. We have integrated the psychological components into an existing crowd simulation system, HiDAC [93].

For instance, for the personality module, we have collected adjectives identifying each personality factor and defined a direct mapping between the parameters in HiDAC and the personality traits. In contrast to the low-level parameter tuning process in previous work, we now let the user choose from higher-level concepts related to human psychology. Thus, the user is freed from understanding the underlying methodologies used in HiDAC. Our mapping also decreases the number of parameters that need to be set from 13 to 5. Using a personality model enabled us to move a user's focus to the character of the agents instead of behavioral parameters while providing us with a somewhat widely accepted structure for describing character. We have evaluated how people perceive the differences of personality through user studies. The results are promising as they indicate high correlation between our parameters and the participants' perception of these parameters.

The second part of our contribution is the simulation of different types of crowds. These crowd types range from audience to mobs. We enable the animator to create various scenarios, giving each agent different roles and personality traits. The agents then act according to the scenario, showing different behaviors based on their personalities, emotions and moods. As well as high level behaviors, they respond with facial and bodily gestures such as changing their posture depending on their current emotional state.

1.3 System Overview

The mind of a virtual agent consists of several components that determine cognitive, perceptual and psychological characteristics. The agent behaves according to the interaction of these features with environmental stimuli. All these components will be detailed in the following chapters. In this chapter, we overview the elements that comprise an agent as shown in Figure 1.1.

The cognitive unit of an agent's mind is the appraisal component. Appraisal determines how agents assess events, other agents, themselves and objects. Their assessment is processed according to decision making strategies and produces emotional outcome. Emotions and intrinsic personality traits affect the mood state. All these psychological components determine the agent's behavior explicitly or implicitly. For instance, facial gestures and postures depend on the emotional state, whereas local motion choices depend on all three components of psychology as well as goals, standards and attitudes.

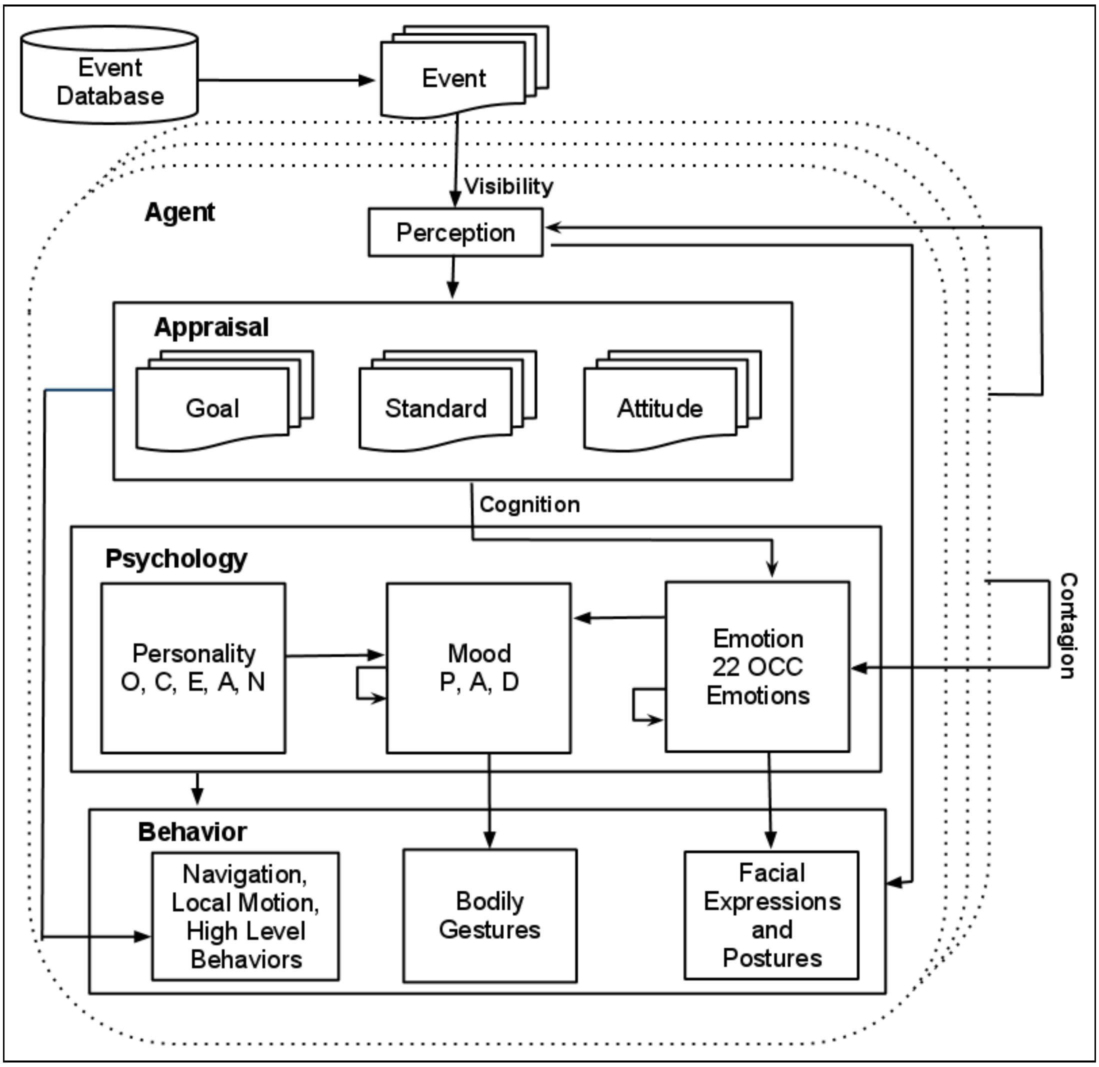


Figure 1.1: System Overview

1.4 Outline of Thesis

The organization of the thesis is as follows: Chapter 2 gives a literature survey on crowd simulation and related fields. Chapter 3 formulates the underlying psychological model. Chapter 4 defines the behavior of virtual crowds based on the classification of crowds. Chapter 5 explains our experiments on validating personality to behavior mapping and presents some visual and runtime performance results. Chapter 6 gives conclusions with possible future work implications.

Navigation is performed by discretizing the environment and computing a cell portal graph. We explain the cell portal graph computation in Appendix A. Finally, we discuss functionality and the user interface of the system in Appendix B.

Chapter 2

Related Work

Computational models are categorized into a hierarchy in the order of their appearance in computer graphics [44, 45, 46]. The earliest models were the *geometric models*. Then, forward and inverse kinematics became widely used, and thus *kinematic models* emerged. The next step was the *physical models*. They are used for animating the physical properties of particles, fluids, solids, gases and deformable solids. However, as a result of the desire to further automate the animation process, *behavioral models* emerged. Behavioral modeling involves self-animating characters that perceive environmental stimuli and give appropriate responses. The highest step in the hierarchy is *cognitive models*, through which autonomous characters can be given goals and react deliberatively as well as reactively. The modeling hierarchy can be seen in Figure 2.1. In this chapter, we explain the current state-of-the-art in behavioral and cognitive models for crowd simulation after giving some definitions about behavioral animation systems.

2.1 Definitions about Behavioral Animation

There are four aspects of behavioral animation techniques [103]:

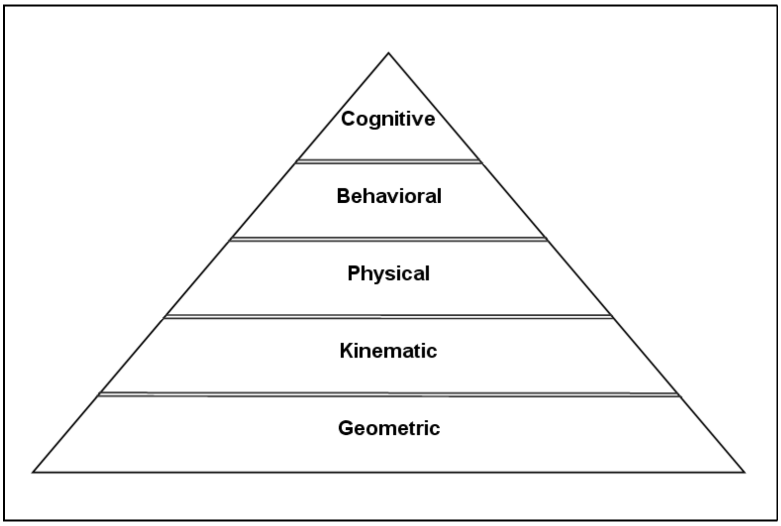


Figure 2.1: Computer graphics modeling hierarchy [46]

1. *Specification and control methods*: Specification can be performed either declaratively or procedurally. Control can be performed either by scripting or sensing the environment.

2. *Generality of the method*: This refers to the type of animations that the technique can generate. For instance, some animation techniques are specific to certain types of behaviors such as flocking.

3. *Directability*: Directability is the degree to which an autonomous character can be externally controlled, which can also be considered the level of autonomy. Considering directability, crowd behavior can be classified as [111]:

    - *Guided crowds*: Behaviors are explicitly defined by the users
    - *Programmed crowds*: Behaviors are programmed in a script language
    - *Autonomous crowds*: Behaviors are specified using rules or complex models

4. *Ease of authoring*: This refers to the types of primitives provided by the system, the user interface and extensibility mechanisms.

In order to realistically simulate virtual characters, we must first understand the basic properties that comprise the characteristics of these agents. A full behavioral animation system should address these issues. These properties can be summarized as follows [111]:

- *Behavior:* Response of an individual, group or species to the environment.
- *Intelligence:* The ability to learn and understand new situations.
- *Autonomy:* The quality or state of self governing.
- *Adaptation:* The ability to survive in unpredictable or dangerous environments.
- *Perception:* Awareness of the elements of the environment through physical sensation.
- *Memory:* The power or process of reproducing or recalling what has been learned and retained especially through associative mechanisms.
- *Emotion:* An affective aspect of consciousness; state of feeling.
- *Consciousness:* The quality or state of being aware especially of something within oneself or the state of being characterized by sensation, emotion, volition, and thought.
- *Freedom:* The extent that the virtual character's future behavior is unpredictable.

Autonomous agents in behavioral animation systems are classified as *situated*, *reactive*, *embodied* and *virtual* [100]. *Situated* agents are located in a virtual world shared by other entities as opposed to *isolated* agents. An agent is *reactive* if it is driven by stimulus and instinctive. On the other hand, an agent is *deliberative* if it is intellectual in the classical artificial intelligence (AI) sense. *Embodied* agents are animated in a physical manifestation such as an autonomous vehicle or a bird. Finally, the term *virtual* is used to discriminate the agents from mechanical robots, which can also be defined as situated, embodied autonomous agents.

Millar et al. classify the components of a behavioral animation system in a generic framework as perception system, behavioral system and motor movement system [83]:

Perception System: Perception techniques determine how an agent perceives its environment and can be classified into three as:

1. *Zonal approach:* This approach involves surrounding the character with perception regions so that any object in this zone can be perceived by the character. The size of the detection zone is important because too small a zone will weaken the collision avoidance and path planning abilities whereas too large a zone will increase the computation time.
2. *Sensory approach:* This approach involves placing synthetic sensors on the character. Different types of sensors for smelling, hearing, seeing etc. can be implemented. The type, location and orientation of each sensor is important for perceiving stimuli from the environment.
3. *Synthetic vision approach:* This approach gives the character a vision of its virtual world. This approach is only useful for vision, no other stimuli will be detected. The advantage of using this method is to learn from research on human vision.

Behavioral System: This system comprises the behavioral basis of animation and it is responsible for the decision making process. Behavior can be either solely reactive as a reflexive response to a stimulus or it can be an intelligent response driven by internal desires and experience of the character. The form of the response is also various. It can be a movement vector as well as a change in the internal attributes. In a fully-implemented system, the behavioral component includes four important modules:

1. state variables including perception variables and mental state,
2. the rule base,
3. the memory module, and

4. the movement module that performs collision handling and path planning.

Different approaches used in behavioral techniques can be classified as:

1. *Behavioral (rule-based) approach:* This approach gives each character a set of rules defining how to react to the environment. It can provide reasonable behaviors in a dynamic environment and it is relatively easy to modify the rules to produce different behaviors. On the other hand, it results in less freedom, i.e., more predictability, it is specific to a particular environment and the number of rules can increase in complex environments.
2. *Network-based approach:* This approach involves creating a series of interconnecting nodes each of which describe the type of behavioral response and these nodes are created as mathematics-based procedures.
3. *Cognitive approach (Artificial intelligence):* This method uses artificial intelligence techniques such as reasoning engines and neural networks to the definition of the behavioral aspects of the animated character. These techniques provide more freedom; however, they are more difficult to control by the animator.
4. *Mathematical approach:* This approach defines the behavior of the characters in mathematical terms. It provides a means of specifying behavioral responses in a precise manner; however, it is not very intuitive for animators.

Motor Movement System: The main functionality of this system is to propel the animated character through its virtual world. Motor movement techniques handle only the movement of the character; path planning is handled by the behavioral component. These techniques actually comprise the animation module of the behavioral animation system. The animated character will receive a movement request from its behavioral component and execute this request by using a specific motor movement approach that will be based on some sort of motion description.

2.2 Behavioral Models

Behavioral models can be categorized into three by considering the possible number of individuals to be simulated, their intelligence level, control mechanisms and collision handling methods. These approaches are particle systems, flocking systems and behavioral systems [91]. Musse et al. extend these categories by adding hierarchical systems [88], which is actually a hybrid of particle, flocking and reactive behaviors. We also include chaos systems, which is a relatively recent approach in behavioral animation techniques.

2.2.1 Particle Systems

Agent-based approaches offer several advantages such as capturing the variability of different individual characteristics and providing heterogeneity to the motion. However, agent-based methods are costly in that each agent must be handled separately, comparing its state with every other agent, thus resulting in $O(n^2)$ time complexity. Several simplifications on agent-based methods have been offered such as local methods, precomputed static plans, global planning on coarse environments and leader-follower models. However, an alternative to agent-based approaches has emerged from the fluid dynamics studies by making an analogy between the crowds and natural phenomena such as the behavior of fluids and gases. Particle systems are composed of many participants with significant dynamics. These systems are physically-based and the control is handled by force fields and global tendency [19, 22, 23]. Although these systems are used to present group and crowd simulations, the individuals in the groups do not have autonomy and heterogeneity.

Hughes introduces a model representing pedestrians as a continuous density field [54]. The model includes an evolving potential function that guides the density field optimally towards its goal. Chenney [26] presents a technique called flow tiles, for representing and designing velocity fields, and gives application examples of crowd simulation on city streets. The most recent work, “continuum crowds", is proposed by Treuille et al. [113], introducing a real-time crowd model based on continuum dynamics. The system is only applicable to large groups with common goals, so individual differences in each group are not handled. The study of continuum crowds is inspired by Hughes, extending it from pure analytical derivations to the simulation of crowds. The authors use a similar potential function to guide pedestrians towards their goal. In addition, it is possible to combine pedestrians into groups and introduce dynamic discomfort fields to handle geographic preferences and obstacles. The continuous equations in the mathematical model are converted into discretizations in time and space. For this purpose, the space is discretized into a regular grid and the physical variables are defined at various locations within each grid cell. The simulation examples demonstrate smooth flow under different conditions and they run at interactive rates.

2.2.2 Flocking Systems

Flocking systems specify animation as distributed global motion with a local tendency. Individuals in flocking systems can seek a goal, move together and avoid collisions. The intelligence level of the individuals of flocks are higher compared to the members of particle systems. Some examples of flocking systems are given in [74, 87].

The principles of behavioral animation are based on the seminal work of Craig Reynolds, who did research on the animation of flocks of birds and schools of fish [98]. Reynolds introduces the term "boid" to refer to bird-like entities, i.e., bird-oids. These entities represent creatures like birds and fish that have flocking or schooling behavior. Each boid acts as an independent actor that maintains proper position and orientation by perceiving the local dynamic environment. The motion of each actor is defined by the laws of simulated physics and a set of programmed behaviors. The main aspect of the system is that the boids have only local information, without knowing the global environment, thus simulating the real-world perception. Each boid perceives its nearby flockmates and the obstacles within its view. The behavior of each individual in the flock is controlled by three simple rules as:

- *collision avoidance*: Avoiding collisions with neighbors,
- *velocity matching*: Tendency to match velocity with neighbors, and
- *flock centering*: Tendency to stay close to neighbors and to be near the center of the flock.

These rules are sorted in the order of decreasing precedence, i.e., collision avoidance has the highest precedence and flock centering has the lowest precedence. Thus, conflicting behaviors are resolved by defining static priorities.

Reynolds extends the technique for flocking to include autonomous reactive behavior. He presents steering behaviors for obstacle avoidance [99] and path determination [100] by introducing constraints. The modeling of autonomous agents is performed in a hierarchical manner and specific emphasis is put on the middle layer of steering. The layers are:

- *action selection*: Strategy, goals and planning,
- *steering*: path determination, and
- *locomotion*: Animation and articulation.

2.2.3 Behavioral Systems

Agents in behavioral systems are more clever compared to the agents in flocking systems. The virtual agents are equipped with synthetic vision and perception of the environment and they are controlled by rules rather than local or global tendencies.

One important study in this field is the simulation of artificial fishes by Terzopoulos et al. [110]. An artificial fish is an autonomous agent that has a three-dimensional, deformable and muscle-based body that conforms with biomechanic and hydrodynamic principles. A fish also has sensors and a brain with motor perception, behavior and learning centers. There are two types of sensors, a temperature sensor that measures the water temperature and a vision sensor that has access to the geometry, material property and illumination information in the rendering pipeline and can identify nearby objects.

The behavior system of an artificial fish is based on intentions. The system runs continuously in a simulation loop, and at each timestep, the intention generator issues an intention based on the habits, mental state and incoming sensory information. The habits are associated with the preferences of the fish on brightness, darkness, cold, warmth, schooling and the gender of the fish. The mental state depends on three variables, which are hunger, libido and fear. The range of each variable determines the urge to eat, mate or avoid danger. The intention generator first checks whether there is an immediate collision. Then, it checks these state variables in the order of fear, hunger and libido and generates a suitable intention at each timestep. If all the state variables are below a certain threshold, the generated intention will be to wander about. The intentions generated influence the behavior routines. There are eight behavior routines: *avoiding-static-obstacle*, *avoiding-fish*, *eating-food*, *mating*, *leaving*, *wandering*, *escaping*, and *schooling*. Dithering is avoided by modeling a short-term memory and persistence is ensured in order to ensure robustness in long duration behaviors such as mating or schooling. Three types of fish are modeled: *predators*, *preys* and *pacifists*.

Blumberg and Galyean [17] combine autonomy with directability. Sometimes it might be necessary to control the animated creature to some extent. In that sense, the study makes three contributions:

1. A control approach that allows an external entity to direct a virtual character at a number of different levels.
2. A general behavioral model for perception and action selection in autonomous animated creatures which also supports external control.
3. A layered architecture that supports extensibility, reusability and multiple

levels of direction.

The modeling of autonomous creatures is performed in a hierarchical manner. The levels in the hierarchy are similar to those of Reynold's [100] and organized in a top-down fashion as follows:

1. Behavior system
2. Motor system
    - Controller
    - Motor skills
    - Degrees-of-freedom
3. Geometric system

Geometric layer portrays the physical attributes of the character, giving its form and appearance. The more complex this layer is, the more sophisticated and expressive characters we can obtain. The second layer, motor system, executes the actions necessary to perform the goals without any knowledge from the environment. This layer acts as an interface between the geometric layer and the behavior layer, supports and provides imperative commands and minimizes the burden on the behavior layer or an external user. Degrees-of-freedom are used to modify the underlying geometry. Motor skills are used to produce more complicated motion such as “walking”. Finally, the controller is used as an abstraction barrier between the behavior system and the underlying motor skills. It maps commands such as “forward”, “turn” or “halt” into calls to turn on or turn off the appropriate motor skill. For instance, “forward” may result in the “walk” motor skill in a dog, or the “move” motor skill in a car. The top level is the behavior layer, which performs the decision making process given the goals and environmental information. It senses the environmental stimuli, chooses the best set of actions for the current state and sends out the necessary signals to the motor control layer.

Behaviors may range from very general to very specific and are organized into groups. External control can be added to the system by changing the motivation or sensor variables of the character or by directly scheduling tasks for execution. All constituent parts of a behavior are accessible during run-time; thus any part can be modified.

External control, i.e., directability, is a feature that has been accepted by many other researchers as well [5, 86, 88, 108]. For instance, Anderson et al. introduce constraints on the individual agents and the entire group [5]. They introduce three types of constraints as: specific agents constrained to pass through a location, the center of mass of the group constrained to a point and the members of the flock constrained to lie within a given shape at a given time. Moreover, Sung et al. define a system where users can dynamically specify the group behaviors at a certain part of the environment by attaching information to the environment [108]. They adopt a two-level scalable approach for the crowd simulation. The higher level uses a situation-based distributed control mechanism that gives each agent the rules about how to react to a specific condition based on the local environment. The lower level uses a probability scheme that computes probabilities over state transitions and then samples to move the simulation forward.

Perlin and Goldberg define a system, *Improv*, based on scripts, which are sets of author-defined rules [97]. The difference of *Improv* from other systems is that it focuses on author's view; it provides tools to create actors that respond to users and other actors in real-time. *Improv* consists of two subsystems: an animation engine and a behavior engine. The animation engine uses procedural techniques to create layered, continuous, non-repetitive motions and smooth transitions between them. The behavior engine, on the other hand, enables authors to create sophisticated rules to govern the way actors communicate, change and make decisions. The animation engine represents the body of the actor whereas the behavior engine represents the mind. The behavior model of *Improv* is similar to that of [17] as it consists of a layered architecture. Information about an actor and his relationship to the environment are stored in actor properties, which describe the aspects of an actor's personality. These properties are specified either when the actor is created or within a clause or script whenever a change is necessary.

2.2.4 Hybrid Systems

Hybrid systems mix particle, flocking and reactive behaviors [111]. The intelligence levels of the agents can vary from none to high in these systems. Musse and Thalmann describe a system called *ViCrowd* that is composed of a hierarchy of virtual crowds, groups and individuals, which constitute the *entities* of the simulation [88]. Individuals are virtual human agents that mimic the behaviors of real humans. Groups refer to a group of agents and crowds refer to a set of groups. Some important concepts about the simulation are *intentions*, *beliefs* and *knowledge*, which are the goals, internal status and the information about the virtual environment of the entities, respectively. Intentions, beliefs, knowledge and perception determine the *crowd behavior*. The system addresses three specific problems:

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, and

3. the required structure to provide interaction with groups of agents during the simulation in real-time.

These problems are solved by considering *crowd structure* and *crowd behavior*. Crowd structure is a hierarchy composed of crowd, groups and agents, where the groups' information is distributed among the individuals. Crowd behavior deals with different levels of autonomy for the individuals. The agents can either act according to specific rules, react to specific events, or can be guided by an interactive process during simulation. Different levels of autonomy has been addressed in [111], as well. This control mechanism also distinguishes hierarchical models from behavioral models.

2.2.5 Chaos Models

Modeling virtual crowds by making use of their chaotic behavior is another method in behavioral approach [53, 101, 107]. As crowds include independently moving individuals, yet exhibit general motion patterns, they can be represented by chaos models. Although these models have only a few parameters, due to the sensitivity of the system to initial conditions and non-regularity, various behaviors can be observed. These methods are superior to using random numbers to achieve variation as these methods are deterministic and it is difficult to create and control general patterns with random numbers. The representation of crowds is at the macro level, contrary to the other micro-level approaches where the focus is on the individuals. Saiwaki et al. [101] state that there are few studies on the behavior of virtual humans with few parameters in contrast to the studies on the behavior of animal groups, because humans demonstrate more complex behaviors.

2.3 Cognitive Models

The techniques introduced up to now are limited in the sense that they do not present any learning ability and confined to pre-specified behaviors. Moreover, they have only behavioral control, which is restricted to decision making. However, cognitive control, which involves reasoning and planning to accomplish long-term tasks is also required in order to achieve full autonomy. Behavioral learning and cognitive models have begun to be explored in computer graphics only recently [16, 25, 28, 29, 30, 46, 84, 112].

Funge introduces cognitive modeling as a further step to behavioral modeling [44, 45, 46]. He defines Cognitive Modeling Language, CML, to specify domain knowledge with terms of actions, their preconditions and their effects, and to direct the character's behavior in terms of goals. Then, the animator only specifies the sketch plan of the animation and the characters take deliberate actions through reasoning to satisfy the plan. Cognitive modeling is decomposed into two subtasks of domain knowledge specification and character direction. Domain knowledge specification is about informing the character about the environment and character direction is about instructing the character to behave in a certain way in order to achieve specific goals. CML provides a high-level interface for description of the desired goals. On the other hand, it can also serve as a traditional programming language, allowing the precise specification of how the character should act. In order to provide simple and powerful semantics for cognitive modeling, situation calculus is used. The syntax of CML employs descriptive keywords with precise mappings to the underlying formal semantics of the situation calculus.

Recently, pedestrian simulation has emerged as a new direction of research in crowd simulation [8, 15]. As well as examining crowd behavior, pedestrian simulation is also important for urban planning [43, 102]. A complex pedestrian animation system, which incorporates perceptual, behavioral and cognitive control components, is introduced as a combination of rule-based and cognitive models [104]. The study treats the crowd from a decentralized point of view, modeling the individuals separately. Individuals are fully autonomous and they perform a rich variety of actions within an urban environment.

2.3.1 Models with Psychological States

Some studies integrate emotions and psychological models and roles into crowd simulation systems and autonomous agents [2, 36, 37, 95, 93, 105, 112]. Silverman et al. describe the PMFServ system that makes use of the psychological elements that affect human behavior [106]. PMFServ is a highly flexible software system that can be utilized in various simulation domains. Although it provides an interface for other cognitive architectures, it is as well a fully functional standalone system to simulate human decision making based on emotions.

Allbeck and Badler give a representational basis for character believability, personality and affect [2]. For this purpose, they describe a Parameterized Action Representation (PAR) that is a representation for the actions as instructions for an agent. PAR allows an agent to act, plan and reason about its behaviors and enables the control of the agent's personality, mood and affect. PAR parameterizes the agent, relevant objects, information about paths, locations, manners and purposes. In order to perform an action, the conditions that specify the action must be satisfied. The agents that execute the action are treated as special objects with their properties stored in a hierarchical database.

Pelechano et al. incorporate psychological models into crowd simulation [95]. Their crowd simulation system deals with the wayfinding process that allows the individuals to explore and learn the internal structure of a building as well as the low-level local motion based on social forces. Thus, the agents can generate a cognitive map for navigation and find their way around an environment about which they have no prior information. The psychological component is included by using PMFServ. Communication and roles are added to achieve individualistic behaviors and spread information about the environment. Individuals have different roles and thus show heterogeneous behavior. The roles depend on two attributes of leadership and training in the existing crowd simulation system. There are trained leaders that have complete knowledge about the environment, untrained leaders and untrained non-leaders, i.e., followers. The agents are thus restricted to only three distinct roles. At this point, the psychological model provides variation through physiology, stress, perception and emotion.

HiDAC [93] is a high density crowd simulation system, which addresses the simulation of local behaviors and global way-finding of crowds in a dynamically changing environment. The behaviors of autonomous agents in HIDAC are governed by the combination of geometrical and psychological rules. Psychological attributes include impatience, panic, and leadership behaviors. Physiological attributes are determined by traits, such as locomotion, energy levels, maximum speed. Agents are provided with skills such as navigation in complex environments, communication, learning, and certain kinds of decision-making. Furthermore, they have perception so that they can react to obstacles, other agents, and dynamic changes in the environment. In order to achieve realistic behavior, collisions are handled both by avoidance and response forces. Over long distances, collision avoidance is applied so that agents can steer around obstacles. Collision response is utilized over shorter distances to prevent agents overlapping with each other and with the environment. In addition to the usual crowd behavior, agents might show pushing behavior or can wait for other agents to pass first depending on their politeness and patience. Pushing behavior arises from varying the personal space threshold of each individual. Impatient agents do not respect others' personal space and they appear to push their way through the crowd. Relaxed agents temporarily stop when another agent moves into their path, while impatient agents do not respond to this feedback and tend to "push".

Another system that involves emotions of virtual agents is presented by Tomlinson and Blumberg [112]. The study is based on social learning for interactive virtual characters, which are wild wolves. Wolves are preferred because of their social similarity to humans and their clear yet complex behaviors in a social group. The system provides a computational model that provides models of learning, emotion and development. Social learning involves the ability to have emotions, to express these emotional states and to remember an association between environmental stimuli and emotional states.

In order to represent individual differences through psychological states, some studies focus on single agents as opposed to crowds. Research on Embodied Conversational Agents (ECAs) introduce agents within different contexts that can communicate with the user through various means. As well as the recognition of social cues, these agents have to present different expressions. Ball and Breese introduce an early work on the modeling of emotions and personality in conversational agents [9]. Virtual characters recognize the user's emotions and personality and give appropriate responses accordingly. Egges et al. study the simulation of the personality, emotions and mood for conversational virtual humans [38]. In addition, Egges et al. present a system that incorporates bodily gestures to virtual humans according to their emotional states [39]. Another system that focuses on conversational agents is introduced by Breitfuss et al. [21]. The system offers methods for using dialogues in text format to simulate conversational agents with eye-gazing behavior and non-verbal gestures. Conversational agents with emotion dynamics are also studied in [12]. The system is composed of three orthogonal axes, which are emotion, mood and boredom.

Gratch and Marsella study how psychological theories of emotion can help the design of autonomous agents by clarifying the interaction between emotion and cognition [51]. Later, they introduce a computational model of emotions, i.e., the EMA model, which stands for Emotion and Appraisal [72, 73]. The model focuses on the dynamics of emotional processes and illustrates how a single-level appraisal model facilitates emotion modeling. Appraisal theories state that emotions are activated through our evaluations of the environment. FLAME is a computational model of emotions, which uses fuzzy logic to map events and expectations to emotions [42]. The model also incorporates machine learning in order for the agents to learn the impacts of events on their goals. Gebhard introduces ALMA - A Layered Model of Affect [47]. ALMA represents three distinct types of affect, i.e., personality, moods and emotions, each of which is related to different human tasks. A later study presents a model that visualizes the affective state of virtual agents by their personality and emotions [6]. Kessler et al. introduce a system called SIMPLEX, which stands for Simulation of Personal Emotion Experience [60]. SIMPLEX is based on the appraisal theory of emotions and it enables the control of multiple virtual agents.

Li et al. propose a framework that uses the OCEAN model of personality to define and formulate a pedagogical agent in a social learning environment [71]. An architecture that combines the bodily emotion dynamics with cognitive appraisal is the WASABI system [13], in which primary and secondary emotions are simulated. Primary emotions are the basic emotions that determine facial expressions, whereas secondary emotions result from reasoning about events based on experiences.

Kasap and Thalmann present a survey about the features that make up intelligent virtual agents [59]. Perception, decision making and personification are among the many characteristics that are mentioned in the survey.

2.3.2 Learning

Learning abilities allow the virtual agents to make decisions according to their experiences by creating a cognitive map of the environment. Most of the systems in the literature use reinforcement learning; thus we will briefly overview the terms and definitions related with this type of learning.

Reinforcement learning is an unsupervised learning technique that can be defined as learning from experience in the absence of a teacher [16]. In this learning technique, the world is taken to be in one of a set of perceivable states. The goal of reinforcement learning is to learn an optimal sequence of actions to take the agent from an arbitrary state to the goal state. The main approach is to probabilistically explore states, actions and their outcomes to learn how to act in a given situation. *State* refers to a specific configuration of the world. The set of all represented configurations of the world is called the *state space.* An agent can change the state of the world by performing an *action.* Each agent is assumed to have a finite set of actions and it can perform only one at a time. A *state-action* pair, $< S/A >$, is a relationship between a state S and an action A. It is typically related with a numerical value like *future expected reward*, which gives the value of performing an action A in a given state S. A *policy* represents the probability with which the agent selects an action at a specific state. When the agent reaches a goal state, it receives a *reward* or *reinforcement.*

The most popular reinforcement learning technique is *Q-Learning* [115]. In Q-Learning, state-action space is stored in a lookup table. Each row represents a state and each column represents an action in the table. An entry in the table represents the *Q-value* of a given state-action pair with respect to getting a reward. The optimal value for each state-action pair can be learned by exhaustive search of the state-action pairs and by a local update rule to reflect the consequences of taking a given action in a given state with respect to achieving the goal state.

An important learning example is given by Blumberg et al. [16], where an autonomous virtual dog is interactively taught to perform a desired behavior.

The system employs reinforcement learning along with learning inspired from animal training, i.e., clicker training. The virtual dog mimics the behavior of a real dog by performing the best action in a given context, assessing the relative reliability of its actions in producing a reward and altering its choice of action accordingly.

Another system that uses reinforcement learning is described by Conde et al. [28]. The system is interesting as it does not use reinforcement learning in its classical approach but as a behavioral engine for exploration, learning and visiting the virtual environment. Thus, the interest is in the learning process itself rather than the optimization of learning. The system makes use of situated AI, which involves adaptive artificial systems evolving in an environment that is not entirely predictable. The autonomous and intelligent agents react to their environment by making decisions based on their perception, memory and logic. Intelligence accounts for the ability to make plans and carry out tasks based on the actual state of the virtual environment. Autonomy refers to the agent's capacity to visit and memorize the given virtual environment without any external intervention.

Conde and Thalmann introduce a new low-level learning technique as an alternative to classical Q-learning [30]. The proposed method uses a tree search algorithm with inverse reinforcement learning. The system's objective is to allow the virtual agent to explore an unknown virtual environment and to build structures in the form of cognitive models or maps. Then, the virtual agent can dissipate this information to other agents. Learning through observation of an expert agent is similar to imitation and called apprenticeship learning. The steps of the learning process are as follows:

1. First, a tree search algorithm A* is used to observe the state sequences generated by the user (expert).
2. Q-decomposition approach that uses all pseudo value function components (vision, avoidance and navigation) is integrated.
3. Apprenticeship learning via inverse reinforcement learning is adapted to the behavioral animation.

2.3.3 Motion and Path Planning for Crowds

In artificial intelligence, planning is related with searching for a sequence of logical operators or actions that transform an initial world state into a desired goal state [66]. Motion planning and path planning problems arise in fields such as robotics, assembly analysis, virtual prototyping, manufacturing and computer animation, but the origin of the problem is in robotics. The main purpose for the object is to plan its own motion. In order to plan a motion, the object must have some knowledge about the environment and find a collision-free path among the obstacles in the environment [1, 32]. The path should be preferably short. A classical motion planning problem is known as the *Piano Mover's Problem*, which is about moving a piano from one room of a house to another without hitting the static obstacles [66].

Detailed surveys on motion planning can be found in Latombe [65], Overmars [90] and Baños et al. [49]. Motion planning for crowd simulation has been studied by many researchers [7, 10, 11, 22, 57, 58, 63, 64, 92]. Motion planning approaches can be classified in three groups as [90, 92] *potential fields*, *cell decomposition methods* and *roadmap methods*.

2.3.3.1 Potential Fields

Potential fields put repulsive powers on the obstacles in the environment and attractive powers on the agent's destination. Thus, the object tries to move in the direction of the goal while being pushed away by obstacles. Due to the use of local properties only, the object may move in the wrong direction, resulting in a deadlock situation; getting trapped in local minima. This approach was first introduced by Khatib [61].

2.3.3.2 Cell Decomposition Methods

Cell decomposition methods divide the free space into a number of discrete cells. These methods either use approximate decomposition [62], in the form of grids or quadtrees, or exact decomposition, in the form of convex cells to cover the entire free space. Convex cells provide constant time to compute a path between any two configurations within a cell.

These algorithms are easy to implement; however, they are ineffective if the resolution is low. Moreover, when the dimension of the configuration space gets higher or when the complexity of the scene is very large, the number of cells required increases too much to be practical.

2.3.3.3 Roadmap Methods

Roadmaps discretize the navigation space in a network of paths made up of lines and curves along which the object can move free of collisions [109]. The roadmap can be considered a graph and thus the problem is reduced to graph searching. The difficulty of these methods is to compute an effective roadmap.

2.4 Evaluation of Crowds

Crowd simulations are normally evaluated subjectively regarding the realism of the simulation. It was not until recently that have more objective methods for evaluation been published. A current study evaluating the perception of pedestrian orientations is conducted by Peters et al. [27]. The work aims at determining the effect of the orientation and context rules for characters in static scenes on perceived plausibility. McDonnell et al. analyze the perceptual impact of the cloning of virtual characters for simulating large crowds [75]. Clones of appearance are found to be easier to recognize than clones of motion; however, clones can be disguised by random orientation and color modulation. The study works as a guide for developers to create realistic looking crowds. Pelechano et al. evaluate how people perceive crowds in virtual environments by means of presence studies [96]. The authors conclude that interaction with the crowd members increases the human subject's sense of presence. Lerner et al. introduce the data driven evaluation of crowds [70]. Their motivation underlies the argument that even though crowd simulations look realistic from a distance, individual behaviors may look odd when examined closely. Therefore, they compare the simulation results with video footage of real crowds using similarity metrics.

2.5 Theories of Crowd Psychology

Since this thesis study is multidisciplinary and aims to combine different aspects of crowd behavior, we need to understand the fundamentals of crowd behavior in order to create realistic simulations. This section reviews the psychology literature on collective behavior.

The very first theory that analyzes collective behavior is the transformation or contagion theory, which is introduced by LeBon [67]. The theory suggests that crowds show mental homogeneity as a result of social contagion. Also, responsibility through anonymity is one of the reasons that causes the crowd to act illogically. Blumer [18] supports the contagion theory by systematizing it. He explains five steps to collective behavior. First an exciting event occurs, drawing the attention of some people. Then, milling behavior emerges as a result of circular reaction. After that, a common object of attention emerges due to milling. Next, social contagion and a common attention object lead to fostering of common impulses. Finally, elementary collective behavior is observed.

Convergence theory states that crowd is made up of individuals having similar behaviors, as opposed to the contagion theory, which states that individuals' behaviors change after the crowd is formed. Allport [3] discusses that individuals make up the crowd and therefore their characteristics determine crowd behavior. For instance, more ignorant people would change their behaviors first in an emergent event. Thus, Allport introduces the predisposition or convergence theory. Milgram [82] and Dollard [35] support the predisposition theory and argue that reward-based learning is applied to crowds and individual responses are intensified in the crowd.

Turner and Killian introduce the emergent-norms theory [114]. According to this theory, unusual collective behavior comes out of new behavioral norms in case of a precipitating event. The theory suggests that collective behavior is not irrational. Turner and Killian indicate that there are five kinds of people involving in a crowd, who are either ego-involved, concerned, insecure, curious or exploiter.

Berk [14] states that crowd behavior derives from game theory and decision theory, where crowd members anticipate reward and support or payoffs. Last but not least, Clark McPhail, in his book "The Myth of the Madding Crowd", reviews theories of crowds from past to present [76] and introduces his own theory composed of individual behavior and control systems theories. He suggests that an individual is composed of thousands of control systems arranged hierarchically.

Chapter 3

Simulation of the Psychological State

In order to simulate human behavior we should first examine the psychological foundations. In this chapter, we explain our computational psychology model and formulate "affect".

Personality, mood and emotion are the three basic aspects of affect. They differ according to their temporal characteristics. Personality is the long term affect. It is intrinsic and it usually does not change over time. Emotions are short-term and they are elicited due to events, other agents or objects [89]. They influence memory, decision making and other cognitive capabilities [20, 41, 55]. Finally, mood is the medium-term affect. Moods last longer than emotions; however they are not as stable as personality. Research shows that moods also have major impact on cognitive functioning [85].

3.1 Personality

Personality is a pattern of behavioral, temperamental, emotional, and mental traits that define an individual. There is still considerable controversy in personality research over how many personality traits there are, but the Five Factor or OCEAN model is popular and it is the one we have chosen for our work [116]. The five factors, which are orthogonal dimensions of the personality space, are openness, conscientiousness, extroversion, agreeableness and neuroticism.

- *Openness* describes a dimension of personality that portrays the imaginative and creative aspect of human character. Appreciation of art, inclination towards going through new experiences and curiosity are characteristics of an open individual.
- *Conscientiousness* determines the extent to which an individual is organized, tidy and careful.
- *Extroversion* is related to how outgoing and sociable a person is.
- *Agreeableness* is a measure of friendliness, generosity and the tendency to get along with other people.
- *Neuroticism* refers to emotional instability and the tendency to experience negative emotions. Neurotic people tend to be too sensitive and they are prone to mood swings.

Each factor is bipolar and composed of several traits, which are essentially the adjectives that are used to describe people [48]. Some of the relevant adjectives describing each of the personality factors for each pole are given in Table 3.1.

The crowd is composed of subgroups with different personalities. Variations in the characteristics of the subgroups influence the emergent crowd behavior. The user can add any number of groups with shared personality traits and can edit these characteristics during the course of the animation. An agent's personality π is a five-dimensional vector, where each dimension is represented by a personality

O+	Curious, alert, informed, perceptive
O-	Simple, narrow, ignorant
C+	Persistent, orderly, predictable, dependable, prompt
C-	Messy, careless, rude, changeable
E+	Social, active, assertive, dominant, energetic
E-	Distant, unsocial, lethargic, vigorless, shy
A+	Cooperative, tolerant, patient, kind
A-	Bossy, negative, contrary, stubborn, harsh
N+	Oversensitive, fearful, dependent, submissive, unconfident
N-	Calm, independent, confident

Table 3.1: Trait-descriptive adjectives

factor, ψ_i. The distribution of the personality factors in a group of individuals is modeled by a Gaussian distribution function N with mean μ_i and standard deviation σ_i:

$$\pi = < \psi_O, \psi_C, \psi_E, \psi_A, \psi_N > \tag{3.1}$$

$$\psi_i = N(\mu_i, \sigma_i^2),\ for\ i \in \{O, C, E, A, N\}, \tag{3.2}$$

where $\mu \in [0, 1]$ and $\sigma \in [-0.1, 0.1]$.

The overall behavior by personality for an individual is a combination of different behaviors. Each behavior is a function of personality as:

$$\beta = (\beta_1, \beta_2, \ldots, \beta_n) \tag{3.3}$$

$$\beta_j = f(n),\ for\ j = 1, \ldots, n \tag{3.4}$$

$$\tag{3.5}$$

Since each factor is bipolar, ψ can take both positive and negative values. For instance, a value of 1 for extroversion means that the individual has extroverted character; whereas a value of -1 means that the individual is highly introverted.

3.1.1 Personality-to-Behavior Mapping

The agents' personality factors (adjectives) are mapped into low-level parameters and the built-in behaviors in the HiDAC model, as shown in Table 3.2. A positive factor takes values in the range $[0.5, 1]$, whereas a negative factor takes values in the range $[0, 0.5)$. A factor given without any sign indicates that both poles apply to that behavior. For instance E+ for a behavior means that only extroversion is related to that behavior; introversion is not applicable. As indicated in Table 3.2, a behavior can be defined by more than one personality dimension. The more adjectives of a certain factor defined for a behavior, the stronger is the impact of that factor on that behavior. Thus, we assign a weight to the factor's impact on a specific behavior. For instance, ω_{EL} is the weight of extroversion on leadership and it takes a value in the range $[0, 1]$. The sum of the weights for a specific type of behavior is 1. Now, we can see how the mapping from a personality dimension to a specific type of behavior is performed. We have defined the behavior parameters for an agent i as follows:

Leadership: Leaders tend to have more confidence in themselves and they help others find their way through a building. They remain calm under emergency situations. Each agent has a leadership percentage determined by its extroversion, and stability. The leadership behavior is computed by:

$$\beta_i^{Leadership} = \omega_{EL}\ \psi_i^E + \omega_{NL}\ (1 - \psi_i^E), \tag{3.6}$$

where $\beta_i^{Leadership} \propto E$ and $\beta_i^{Leadership} \propto^{-1} N$, and $\beta_i^{Leadership} \in [0, 1]$.

Trained: Trained agents have complete knowledge about the environment. Since being trained requires curiosity and trained people are informed, this parameter is associated with openness. Being trained is a Boolean parameter, and therefore, it is represented by a probability function. As openness increases, the probability that the agent is trained increases as:

Leadership	Dominant, assertive, bossy, dependable, confident, unconfident, submissive, dependent, social, unsocial	E, A-, C+, N
Trained/not trained	Informed, ignorant	O
Communication	Social, unsocial	E
Panic	Oversensitive, fearful, calm, orderly, predictable	N, C+
Impatience	Rude, assertive, patient, stubborn, tolerant, orderly	E+, C, A
Pushing	Rude, kind, harsh, assertive, shy	A, E
Right preference	Cooperative, predictable, negative, contrary, changeable	A, C
Avoidance /personal space	Social, distant	E
Waiting radius	Tolerant, patient, negative	A
Waiting timer	Kind, patient, negative	A
Exploring environment	Curious, narrow	O
Walking speed	Energetic, lethargic, vigorless	E
Gesturing	Social, unsocial, shy, energetic, lethargic	E

Table 3.2: Low-level parameters vs. trait-descriptive adjectives

$$P_i(Trained) = \omega_i^O \tag{3.7}$$

$$\beta_i^{Trained} = \begin{cases} 0 & \text{if } P_i(Trained) \geq 0.5 \\ 1 & \text{otherwise} \end{cases} \tag{3.8}$$

where $P_i(Trained) \propto O$ and $\beta_i^{Trained} \in \{0, 1\}$.

Communication: This parameter determines whether the agents communicate with each other to give information about the explored areas during a building evacuation. Similar to being trained, communication depends on the probability of agent behavior. As extroversion increases, the probability that the agent communicates increases as:

$$P_i(Communication) = \psi_i^E \tag{3.9}$$

$$\beta_i^{Communication} = \begin{cases} 0 & \text{if } P_i(Communication) \geq 0.5 \\ 1 & \text{otherwise} \end{cases} \tag{3.10}$$

where $P_i(Communication) \propto E$ and $\beta_i^{Communication} \in \{0, 1\}$.

Panic: Under emergency situations, agents show panic behavior depending on their stability and conscientiousness traits. When they panic, their walking speed increases and they do not respect waiting rules.

$$\beta_i^{Panic} = \omega_{NP}\ \psi_i^N + \omega_{CP}\ f(\psi_i^C) \tag{3.11}$$

$$f(\psi_i^C) = \begin{cases} -2\psi_i^C + 2 & \text{if } \psi_i^C \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

where $\beta_i^{Panic} \propto N$ and $\beta_i^{Panic} \propto^{-1} C+$, and $\beta_i^{Panic} \in [0, 1]$.

Impatience: The impatience parameter is implemented by dynamically modifying the route selection based on environmental changes. It depends on the politeness and assertiveness of an agent.

$$\beta_i^{Impatience} = \omega_{EI}\ f(\psi_i^E) + \omega_{AI}\ (1-\psi_i^A) + \omega_{CI}\ (1-\psi_i^C) \quad (3.13)$$

$$f(\psi_i^E) = \begin{cases} -2\psi_i^E - 1 & \text{if } \psi_i^E \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.14)$$

where $\beta_i^{Impatience} \propto E+$ and $\beta_i^{Impatience} \propto^{-1} A, C$, and $\beta_i^{Impatience} \in [0,1]$.

Pushing: HiDAC can realistically simulate an individual's respect for others: an agent can try to force its way through a crowd by pushing others, exhibit more respectful behavior when desired, make decisions about letting others walk first, and queuing when necessary. Disagreeable agents tend to push others more as they are harsh and impolite. Similarly, extroverted agents show pushing behavior as they tend to be assertive.

$$P_i(Pushing) = \omega_{EP}\ \psi_i^E + \omega_{AP}\ (1-\psi_i^A) \quad (3.15)$$

$$\beta_i^{Pushing} = \begin{cases} 1 & \text{if } P_i(Pushing) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (3.16)$$

where $P_i(Pushing) \propto E$, $P_i(Pushing) \propto^{-1} A$ and $\beta_i^{Pushing} \in \{0,1\}$.

Right preference: When the crowd is dispersed, individuals tend to look for avoidance from far away and they prefer to move towards the right hand side of the obstacle they are about to face. This behavior shows the individual's level of conformity to the rules. The right preference behavior is a probability function. If an agent is disagreeable or non-conscientious, then that agent can make right or left preference with equal probability. On the other hand, an agent prefers the right side by increasing probability proportional to its agreeableness and conscientiousness values if these are positive.

$$P_i(Right) = \begin{cases} 0.5 & \text{if } \psi_i^A < 0 \text{ } or \text{ } \psi_i^C < 0 \\ \omega_{AR}\psi_i^A + \omega_{CR}\psi_i^C & \text{otherwise} \end{cases} \tag{3.17}$$

$$\beta_i^{Right} = \begin{cases} 1 & \text{if } P_i(Right) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

where $P_i(Right) \propto A, C$ and $\beta_i^{Right} \in \{0, 1\}$.

Personal space: Personal space determines the territory in which an individual feels comfortable. Agents try to preserve their personal space when they approach other agents and when other agents are approaching from behind. However, these two values are not the same. According to the research on Western cultures, the average personal space of an individual is found to be 0.7 meters in front and 0.4 meters behind [52]. The personal space of an agent i with respect to another agent j is thus:

$$\beta_{i,j}^{PersonalSpace} = \begin{cases} 0.8 \ f(i,j) & \text{if } \psi_i^E \in [0, \frac{1}{3}) \\ 0.7 \ f(i,j) & \text{if } \psi_i^E \in [\frac{1}{3}, \frac{2}{3}] \\ 0.5 \ f(i,j) & \text{if } \psi_i^E \in (\frac{2}{3}, 1] \end{cases} \tag{3.19}$$

$$f(i,j) = \begin{cases} 1 & \text{if } i \text{ } is \text{ } behind \text{ } j \\ \frac{0.4}{0.7} & \text{otherwise} \end{cases} \tag{3.20}$$

where $\beta_i^{PersonalSpace} \propto^{-1} E$ and $\beta_i^{PersonalSpace} \in \{0.5, 0.7, 0.8\}$.

Waiting radius: In an organized situation, individuals tend to wait for space available before moving. This waiting space is called the waiting radius and it depends on the kindness and consideration of an individual, i.e., the agreeableness dimension.

$$\beta_{i,j}^{WaitingRadius} = \begin{cases} 0.25 & \text{if } \psi_i^A \in [0, \frac{1}{3}) \\ 0.45 & \text{if } \psi_i^A \in [\frac{1}{3}, \frac{2}{3}] \\ 0.65 & \text{if } \psi_i^A \in (\frac{2}{3}, 1] \end{cases} \tag{3.21}$$

where $\beta_i^{WaitingRadius} \propto A$ and $\beta_i^{WaitingRadius} \in \{0.25, 0.45, 0.65\}$.

Waiting timer: If two individuals are heading to the same direction, they wait for the other to move first. The time they wait, i.e. the duration that they show patience towards the other, depends on their agreeableness.

$$\beta_{i,j}^{WaitingTimer} = \begin{cases} 1 & \text{if } \psi_i^A \in [0, \frac{1}{3}) \\ 5 & \text{if } \psi_i^A \in [\frac{1}{3}, \frac{2}{3}] \\ 50 & \text{if } \psi_i^A \in (\frac{2}{3}, 1] \end{cases} \tag{3.22}$$

where $\beta_i^{WaitingTimer} \propto A$ and $\beta_i^{WaitingTimer} \in \{1, 5, 50\}$.

Exploring the environment: Individuals are assigned specific behaviors to perform. The number of actions they complete depends on their curiosity. Open people are more likely to explore different experiences, and hence, perform more actions. The openness factor determines the time an individual spends on exploring the environment. Thus, the number of actions that an individual completes increases by the degree of openness.

$$\beta_i^{Exploring} = 10\psi_i^O, \tag{3.23}$$

where $\beta_i^{Exploring} \propto O$ and $\beta_i^{Exploring} \in [0, 10]$.

Walking speed: The maximum walking speed is determined by an individual's energy level. As extroverts tend to be more energetic while introverts are more lethargic, this parameter is controlled by the extroversion trait.

$$\beta_i^{WalkingSpeed} = \psi_i^E + 1, \tag{3.24}$$

where $\beta_i^{WalkingSpeed} \propto E$ and $\beta_i^{WalkingSpeed} \in [1, 2]$.

Gesturing: The amount of gestures used during a conversation is a sign of how sociable a person is. Outgoing people use more gestures than shy people, which is an indication of extroversion.

$$\beta_i^{Gesturing} = 10\psi_i^E, \tag{3.25}$$

where $\beta_i^{Gesturing} \propto E$ and $\beta_i^{Gesturing} \in [0, 10]$.

3.2 Emotion

Since the effect of mood and emotion on behavior is not as straightforward as the personality-to-behavior mapping, we postpone the explanation of our mapping to the next chapter. Mood and emotion combined with external stimuli determine the type of bodily gestures and certain navigational preferences since humans generally act based on the context.

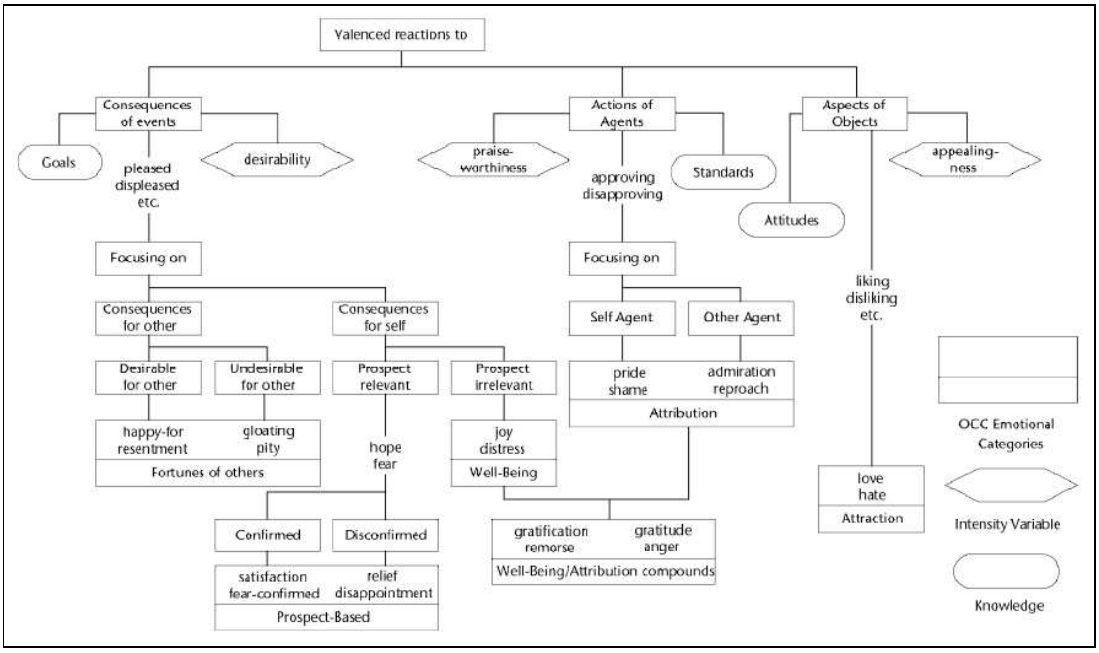


Figure 3.1: The OCC Model (Reprinted from [89])

Emotions take values between 0 and 1. An emotion is active if it has a value different from 0. As the OCC Model suggests, activation of an emotion depends on the context. In the next chapter, after describing different scenarios, we explain how each emotion is activated by environmental stimuli.

Empathy is another factor that affects the emotional state in addition to goals, standards and attitudes. The emotional state is computed as:

$$e_t = \textit{f(goals, standards, attitudes)} + \lambda(\varepsilon), \tag{3.26}$$

where λ is a function of empathy ε. Before we explain the computation of empathy and λ, we should elaborate on the emotional state. An emotion is not forever active; it decays over time. At each timestep, the emotion value is decreased as:

$$e_t = e_{t-1} - \beta e_{t-1}, \tag{3.27}$$

The variable β determines the speed of emotional decay and it is proportional to neuroticism as in the case of mood decay.

When an emotion is activated, it affects certain behaviors. Humans' emotions and attitudes can be inferred from their nonverbal behaviors [40, 50] such as their postures, gestures and facial expressions. Although the OCC model highly covers the emotion space, finding a mapping between the OCC emotions and facial expressions is not straightforward. Ekman studied the facial expressions of emotions [41] and defined six types of emotions, which are happiness, sadness, anger, fear, disgust and surprise. Since we basically implement the OCC emotions, we define a correspondence between Ekman emotions and OCC emotions as follows:

- *Happiness*: HappyFor, Gloating, Gratification, Joy, Pride, Admiration, Love, Satisfaction, Relief.
- *Sadness*: Disappointment, Distress, Pity, Remorse, Resentment, Shame.
- *Anger*: Anger, Hate, Reproach.
- *Fear*: Fear, FearsConfirmed.
- *Disgust*: Hate, Reproach.
- *Surprise*: -.

There is no correlating emotion for surprise since it is not considered to have a cognitive basis. In addition, hate and reproach are mapped to both anger and disgust. Thus, the mapping is not straightforward; we need to make an inference from the context.

In addition to facial expressions, body postures depend on the emotional state as well [31]. We attribute the same six Ekman emotions to static body postures. For instance, happy people tend to have a straight posture with high shoulders, looking more confident. In contrast, sad people have collapsed upper bodies with low shoulders, looking downwards. We constructed the meshes for these postures and facial expressions offline. Moreover, we designed 10 different gestures to visualize the reactions of agents. Figure 3.2 shows these bodily gestures incorporated to our system.

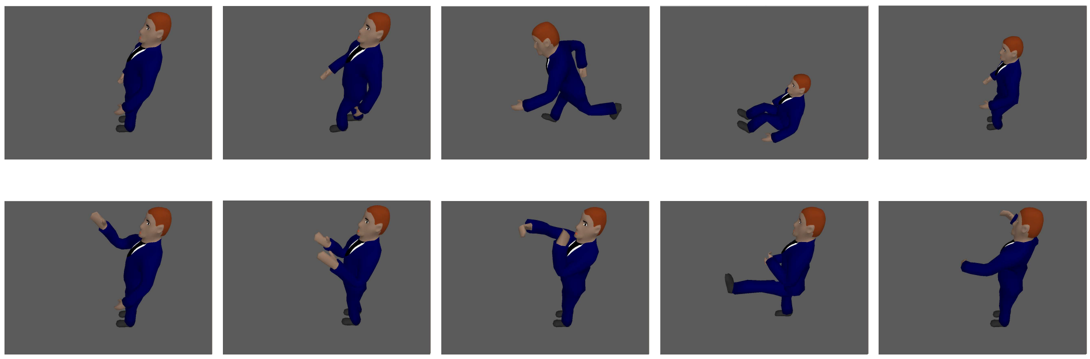


Figure 3.2: Gestures from left to right and top down: Standing, walking, running, sitting, jumping, waving, applauding, punching, kicking, throwing

3.2.1 Emotion Contagion

In its general sense, contagion means the communication of any influence between individuals. It can refer to biological contagion, such as contracting infectious diseases or social contagion, which spans a wide range of areas from economic trends to rumor spreading and thereby resulting in collective behavior. We incorporate a social contagion model into our system in order to simulate the spread of emotions. For this purpose, we follow the approach proposed by Dodds and Watts [33, 34]. The model is a threshold model as opposed to independent interaction models, in which successive contacts may result in contagion with independent probability. Threshold models, on the other hand, suggest that the probability of contracting infection increases as individuals get exposed to infected individuals.

The model states that, in a population, individuals can be in one of the two states: susceptible or infected. These terms are derived from biological contagion; however, they are also meaningful in a social context. A susceptible individual can be "uninformed" about rumors, or a "non-adopter", in terms of emotional responses. Similarly, an infected individual relates to an "informed" individual, or an "adopter", who adopts the emotional states of other individuals. When susceptible individuals come into contact with the infected ones, they can become infected with some probability. The formal definition is as follows:

When an infected individual i makes contact with a susceptible individual j, j becomes exposed and may get infected with some probability. Exposure means receiving a random dose d_j from a specified probability distribution. All individuals keep a memory of their previous k doses as:

$$D_j(t) = \sum_{t'=t-k+1}^{t} d_i(t') \tag{3.28}$$

If the cumulative dose $D_j(t)$ extends a specified threshold T_j at any time of the simulation, then the individual j becomes infected.

Both the dose and the threshold distributions are log-normal distributions $Log - \mathcal{N}$ with means μ_{dj}, μ_{Tj} and standard deviations σ_{dj}, σ_{Tj}, respectively:

$$\begin{aligned} d_j &= \log -\mathcal{N}(\mu_{dj}, {\sigma_{dj}}^2) && (3.29) \\ T_j &= \log -\mathcal{N}(\mu_{Tj}, {\sigma_{Tj}}^2) && (3.30) \end{aligned}$$

The experience of another's emotions through emotional contagion is the basis of empathy and it leads to imitation of behavior. Empathy is found to be positively correlated with all the five factors of personality [56]. Based on the research done by Jolliffe and Farrington, the correlation values between basic empathy scale (BES) and personality factors are shown in Table 3.3.

Personality	Male	Female
O	0.34	0.15
C	0.17	0.01
E	0.13	0.09
A	0.3	0.24
N	0.02	0.16

Table 3.3: Correlation of the BES to OCEAN factors

Empathy ε takes a value between 0 and 1 and it is computed for a male agent i as follows:

$$\varepsilon_i = \psi_i^O\ 0.34 + \psi_i^C\ 0.17 + \psi_i^E\ 0.13 + \psi_i^A\ 0.3 + \psi_i^N\ 0.02; \tag{3.31}$$

$\lambda(\varepsilon)$ function, which determines how emotions are contracted among humans, is computed as:

$$\begin{aligned} T_i(t) &= \log -\mathcal{N}(\frac{1}{\varepsilon_i}, {\sigma_{T_i}}^2) && (3.32) \\ \lambda_i(t) &= \begin{cases} 1 & \text{if } D_i(t) > T_i(t) \\ 0 & \text{otherwise} \end{cases} && (3.33) \end{aligned}$$

The dose threshold is a function of $\frac{1}{\varepsilon_i}$, because the more empathetic a person is, the more susceptible s—he becomes to the emotions of other people. In order to provide heterogeneity within the crowd, each individual should be susceptible in different levels. These correlation values show us a way to determine the dose and threshold distribution values.

3.3 Mood

We utilize the PAD temperament model in our system [77, 79, 80, 81]. PAD stands for Pleasure-Arousal-Dominance and refers to the three orthogonal scales used to assess emotional predispositions. Mehrabian defines temperament or mood as the average emotional state across a representative sample of life situations [79]. The three traits of mood are found to be nearly orthogonal to each other. Three orthogonal axes ranging from -1 to 1 describe each mood state. Pleasure defines the relative predominance of negative versus positive affective states. Arousal is a measure of how easily a person can be aroused by complex, changing or unexpected information. Finally, dominance determines a person's inclination of controlling and influencing his/her own life versus feelings of being controlled by others. Table 3.4 shows the trait names for all the eight P, A, D quadrants. In that sense, mood is continuous in a three-dimensional space.

+P +A +D Exuberant	-P +A +D Hostile
+P +A D Dependent	-P +A -D Anxious
+P A +D Relaxed	-P -A +D Disdainful
+P A D Docile	-P -A -D Bored

Table 3.4: Mood quadrants

Mood is represented as a three-dimensional vector m_t where the three dimensions refer to P, A and D, respectively. Mood is updated according to emotional state. We follow the ALMA [47] approach for human-like mood changes. Table 3.5 shows the mapping between OCC emotions and mood traits. According to the table, $\mathbf{C_{ij}}$, for $i = 1, \ldots, 22$ and $j = 1, \ldots, 3$ gives the emotion constants for all the 22 OCC emotions with respect to P, A and D values, respectively.

Emotion	**P**	**A**	**D**
Admiration	0.5	0.3	-0.2
Hope	0.2	0.2	-0.1
Anger	-0.51	0.59	0.25
Joy	0.4	0.2	0.1
Disappointment	-0.3	0.1	-0.4
Love	0.3	0.1	0.2
Distress	-0.4	-0.2	-0.5
Pity	-0.4	-0.2	-0.5
Fear	-0.64	0.60	-0.43
Pride	0.4	0.3	0.3
FearsConfirmed	-0.5	-0.3	-0.7
Relief	0.2	-0.3	0.4
Gloating	0.3	-0.3	-0.1
Remorse	-0.3	0.1	-0.6
Gratification	0.6	0.5	0.4
Reproach	-0.3	-0.1	0.4
Gratitude	0.4	0.2	-0.3
Resentment	-0.2	-0.3	-0.2
HappyFor	0.4	0.2	0.2
Satisfaction	0.3	-0.2	0.4
Hate	-0.6	0.6	0.3
Shame	-0.3	0.1	-0.6

Table 3.5: Mapping between OCC emotions and PAD space

We first compute the mood values that correspond to the emotions as the emotion center, ec by following Table 3.5 as:

$$ec_t = \frac{e_t \bullet \mathbf{C}}{||e_t||}, \tag{3.34}$$

where e_t is a 22 dimensional vector corresponding to the OCC emotions.

In order to update the mood, we first find where the current mood m_t stands considering the default mood m_0 and the emotion center ec_t. If it is between m_0 and ec_t, it is pulled towards ec_t. On the other hand, if it is beyond ec_t, it is pushed further from ec_t, meaning that the current mood is boosted by the experienced emotions.

$$m_t = \begin{cases} -c\,\frac{ec_t - m_t}{||ec_t - m_t||} & \text{if } |ec_t - m_t| \bullet |m_0 - m_t| > 0 \wedge |m_t - ec_t| \bullet |m_0 - ec_t| < 0 \\ c\,\frac{ec_t - m_t}{||ec_t - m_t||} & \text{otherwise} \end{cases} \tag{3.35}$$

where the constant c determines the speed of mood update. We compute the default mood m_0 according to personality, for which we use the mapping between the big five factors of personality and mood as given by Mehrabian [78].

$$m_0 = \mathbf{M}\,\pi, \tag{3.36}$$

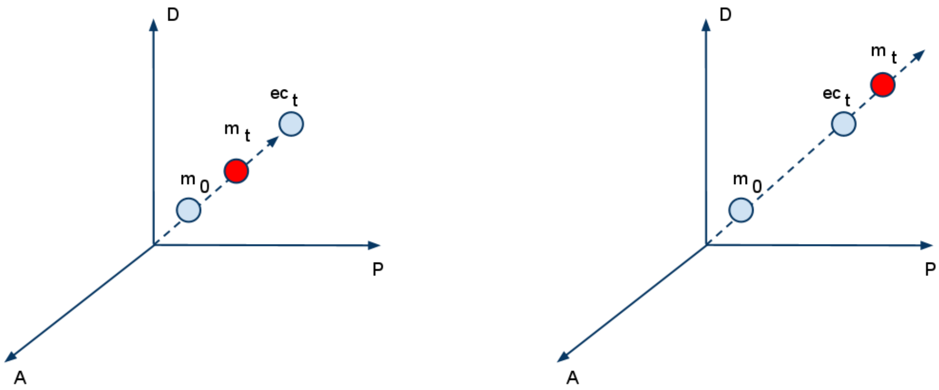
where π is the personality vector $< \psi_O, \psi_C, \psi_E, \psi_A, \psi_N >$ and $\mathbf{M}$ is a constant matrix as:

$$\mathbf{M} = \begin{bmatrix} 0.00 & 0.00 & 0.21 & 0.59 & 0.19 \\ 0.15 & 0.00 & 0.00 & 0.30 & -0.57 \\ 0.25 & 0.17 & 0.00 & -0.32 & 0.00 \end{bmatrix} \tag{3.37}$$

Unlike emotions, moods are more stable in a humans life. However, they decay over time as well; only it takes much longer time than emotional decay. Mood decay is computed as:

$$m_t = m_{t-1} - \alpha(m_0 - m_{t-1}), \tag{3.38}$$

where α is a mood decay variable proportional to neuroticism, since neurotic people tend to experience frequent mood swings. Figure 3.3 shows how the current mood is updated by push and pull phases.



(a) (b)

Figure 3.3: Mood update by (a) pulling towards ec_t and (b) pushing away from ec_t

Since the effect of mood and emotion on behavior is not as straightforward as the personality-to-behavior mapping, we postpone the explanation of our mapping to the next chapter. Mood and emotion combined with external stimuli determine the type of bodily gestures and certain navigational preferences since humans generally act based on the context.

Chapter 4

Crowd Types

In his prominent article, R. W. Brown uses the term collectivity for two or more people who can be discussed as a category [24]. He defines crowds as collectivities that congregate on a temporary basis. Since the reasons that bring crowd members together are various, Brown classifies them in terms of the dominant crowd behavior. He gives a detailed taxonomy of crowds, but basically, he classifies them into two: mobs and audiences. Audiences are passive crowds, who congregate in order to be affected or directed, not to act. Mobs, on the other hand, are active crowds. In fact, the word mob is derived from the word "mobile". There are different tendencies among mobs and audiences. Figure 4.1 shows Brown's taxonomy of crowds.

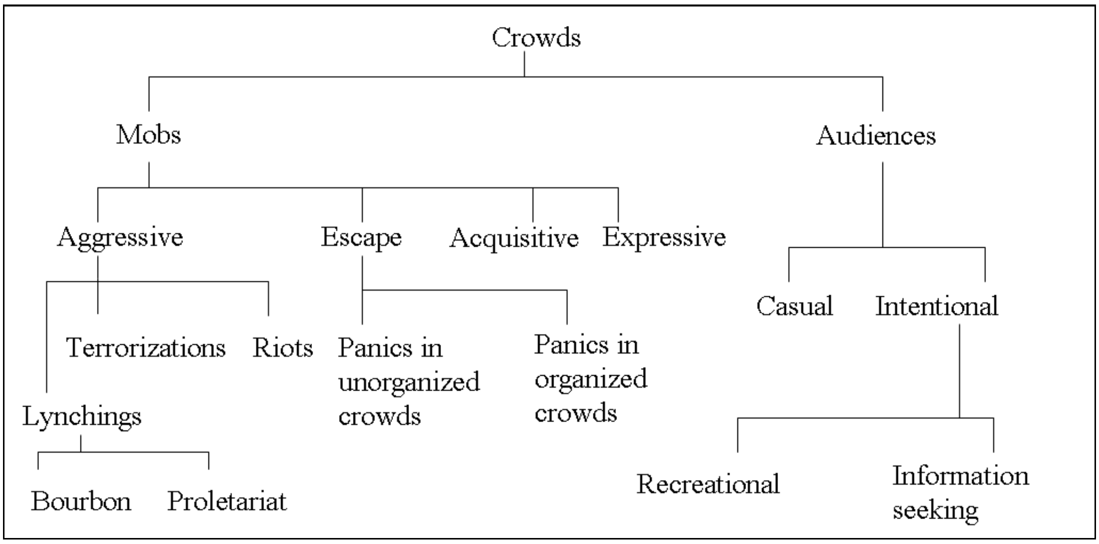


Figure 4.1: Brown's taxonomy of crowd types [24]

According to the classification, mobs are further divided into four groups. They can be aggressive, escape, acquisitive or expressive crowds. It is not always clear into which category a disturbance falls. Aggressive mobs are defined by anger. Lynchings are directed against individuals, whereas terrorizations are directed against groups. Riots are directed against a collectivity and they are urban as opposed to lynchings and terrorizations, which are rural disturbances. Escape crowds are defined by fear. They are panicking crowds, which can be unorganized or organized as in armies. Acquisitive mobs are centripetal and they converge upon a desired object. For example, hunger riots, looting shops and houses are all performed by acquisitive mobs. Finally, expressive mobs congregate for expressing a purpose, such as strikes, rallies, festivals or parades. Similar to mobs, audiences are also classified further. Casual audiences are groups of people who temporarily become polarized through their interest in an event. People gathering around an interest point out of curiosity is an example of casual audiences. Intentional audiences can be either recreational or information seeking. People in a movie theater are examples of recreational audiences whereas people attending classes are examples of information seeking audiences.

We build our system based on a simplified version of this taxonomy. The author can create a scenario and observe the formation of different types of crowds depending on external stimuli and agent roles. External stimuli consist of different types of events, which are:

- attacking → Leading to aggressive mobs,
- explosions → Leading to escape mobs,
- festival → Leading to expressive mobs,
- protest → Leading to expressive mobs, and
- sales → Leading to acquisitive mobs.

As well as emergent events, agents can also have different roles that lead to the formation of different crowd types. These roles are:

- attacker → Leading to aggressive mobs,
- victim → Leading to aggressive mobs,
- provocateur → Leading to aggressive mobs,
- protester → Leading to expressive mobs,
- leader → Leading to expressive mobs,
- audience → Corresponding to casual audiences and may be leading to aggressive, expressive or escape mobs,
- singer → Part of expressive mobs, and
- security → Part of any type of mobs.

Events have both physical and psychological implications on agents. For instance, a virtual human runs away from an explosion and expresses fearful gestures at the same time. In this chapter we will explain different scenarios in detail. These scenarios are explosions, festival, sales and protest.

4.1 State Update

At each time step, the psychological state of the agent is updated first, followed by the computation of physical and cognitive responses. Algorithm 1 shows the state update of an agent.

Algorithm 1: UpdateStep: state update of an agent

```
ComputeEffectsOfEvents();
appraisal.ComputeEventFactor();
ComputeEmotionContagion();
emotionModel.ComputeEmotionalState(appraisal.GetEventFactor());
emotionModel.ComputeMoodState(appraisal.GetEventFactor());
f_NextStep ⇐ PlanNextStep(otherHumans);
//Computed as part of HiDAC, modified slightly
ComputeNextStep(f_NextStep);
```

The procedure "ComputeEffectsOfEvents()" depends on the event type and is explained in the sequel. The procedure computes the effect of the event on the agent depending on its type, location and the agent's role in the event. "ComputeEventFactor()" procedure simply walks down the branches of the OCC decision tree for emotions and updates the corresponding emotion value according to the active goals, standards and attitudes. This procedure and the computation of emotion contagion, emotional and mood states are explained in Chapter 3.

"ComputeNextStep()" is a procedure defined within the scope of HiDAC. It normally computes and sums up all the forces acting on the agent as:

$$f_{Total} = f_{Attrac}\omega_{Attrac} + f_{Walls}\omega_{Walls} + f_{Obsts}\omega_{Obsts} + f_{OtherAgents}\omega_{OtherAgents}, \quad (4.1)$$

where f_{Attrac} is the force towards the attractor position, f_{Walls} is the avoidance force from walls, f_{Obsts} is the avoidance force from obstacles, and $f_{OtherAgents}$ is the avoidance force from other agents. ωs are the corresponding weights for each force. We extend this equation by including forces from attractive and repulsive events (Equation 4.2).

$$
\begin{aligned}
f_{Total} = \; & f_{Attrac}\omega_{Attrac} + f_{Walls}\omega_{Walls} + \\
& f_{Obsts}\omega_{Obsts} + f_{OtherAgents}\omega_{OtherAgents} + \\
& f_{AttracToEvents}\omega_{AttracToEvents} + f_{RepulsionFromEvents}\omega_{RepulsionFromEvents} + \\
& f_{NextStep}\omega_{NextStep}
\end{aligned}
\tag{4.2}
$$

Attractive events are discriminated by their pleasant nature. Agents tend to move towards the location of the attractive event. On the other hand, an explosion, for example, is considered a repulsive event. Agents run away from the explosion region. In addition, agents may have different motivations and therefore different attraction points. For instance, a hostile agent with an intention to attack a victim will be attracted towards the victim. In contrast, the victim will try to elude the attacker. In such cases, $f_{NextStep}$ determines the different forces acting on agents. The procedures for computing the attraction and repulsion forces for events are given in Algorithms 2 and 3.

Algorithm 2: AttractionToEvents: computing the attraction forces for events

```
Output: f_AttracToEvents
Priority p ⇐ 0;
f_AttracToEvents ⇐ 0;
foreach g ∈ Goals do
    if ConseqForSelf(g) ∧ ProspectRelevant(g) ∧ Unconfirmed(g) ∧ Pleased(g) ∧
    GetPriority(p) > p then
        p ⇐ GetPriority(g);
        if GetCell(pos) = GetCell(g.pos) then
            //If agent is in the same cell as goal
            dir ⇐ g.pos − pos;
        else
            dir ⇐ NextAttractorTo(g.pos) − pos;
        f_AttracToEvents ⇐ dir / ||dir||;
if ||dir|| < ϵ then
    //Stop there
    speed ⇐ 0;
```

If the agent and its goal are in the same cell, then the agent can go directly towards the goal. However, if they are in different cells, the "*NextAttractorTo*" method performs path planning to find which portal the agent needs to cross first in order to get to the attraction point.

Algorithm 3: RepulsionFromEvents: computing the repulsion forces for events

---

Output: $f_{RepulsionFromEvents}$

$f_{RepulsionFromEvents} \Leftarrow 0$;
$dir \Leftarrow 0$;
$cnt \Leftarrow 0$;
foreach $g \in Goals$ **do**
  if $ConseqForSelf(g) \ \wedge \ Displeased(g)$ **then**
    $dir \Leftarrow dir + g.pos$;
    $cnt \Leftarrow cnt + 1$;
foreach $a \in Attitudes$ **do**
  if $Disliking(a)$ **then**
    $dir \Leftarrow dir + a.pos$;
    $cnt \Leftarrow cnt + 1$;
if $cnt > 0$ **then**
  $dir \Leftarrow pos - \frac{dir}{cnt}$;
  $f_{RepulsionFromEvents} \Leftarrow \frac{dir}{\|dir\|}$;

Repulsion force is computed by finding a vector oriented away from the center of repulsive events' locations. For convenience, each repulsive event is considered equally strong.

4.2 Expressive Mobs

We examine two types of expressive mobs. The first one is a festival scenario, where agents have fun and the dominant emotion is joy. The second one is a protest scenario with angry agents rallying and conflicting with the security staff.

4.2.1 Festival

The festival event consists of a street concert, where audiences become polarized towards the singer on stage. As well as the audience, there are also provocateurs, who have the purpose of starting fights with audiences. In case of a festival, as we walk down the branches of the decision tree for OCC emotions, the following emotions are triggered for each agent role:

Role: `Audience`
Goal: `Find a place to listen to the singer.`
  State: `Walking`
    `Goals` → `Consequences for self` → `Prospect relevant` →
    `Unconfirmed` → `Pleased` → `Hope`
  State: `Found a place`
    `Goals` → `Consequences for self` → `Prospect relevant` →
    `Confirmed` → `Pleased` → `Satisfaction`
  State: `Found no place`
    `Goals` → `Consequences for self` → `Prospect relevant` →
    `Disconfirmed` → `Pleased` → `Disappointment`
Goal: `Enjoy the concert`
  State: `Waving` ∨ `Jumping` ∨ `Applauding`
    `Goals` → `Consequences for self` → `Prospect irrelevant` →
    `Pleased` → `Joy`
Goal: `Defend against an attacking provocateur`
  State: `Fighting`
    `Goals` → `Consequences for self` → `Prospect irrelevant` →
    `Displeased` → `Distress`
    `Goals` → `Consequences for other` → `Desirable for other` →
    `Displeased` → `Resentment`
Standard: `Provocateurs`
  State: `Fighting`
    `Standards` → `Focusing on other` → `Disapproving` → `Reproach`
Compound Emotion: `Distress + Reproach = Anger`
Standard: `Singer`
  State: `Any`
    `Standards` → `Focusing on other` → `Approving` → `Admiration`

<u>Role:</u> <u>`Provocateur`</u>
Goal: `Provoke fight`
  State: `Fighting`
    `Goals` $\rightarrow$ `Consequences for other` $\rightarrow$ `Prospect relevant` $\rightarrow$
    `Undesirable for other` $\rightarrow$ `Pleased` $\rightarrow$ `Gloating`
Standard: `Audiences`
  State: `Any`
    `Standards` $\rightarrow$ `Focusing on other` $\rightarrow$ `Disapproving` $\rightarrow$ `Reproach`

Algorithm 4 shows how the state transitions are applied in a festival. This procedure is part of the aforementioned method "*ComputeEffectsOfEvents*".

Algorithm 4: ComputeFestivalEffect: application of state transitions in a festival

```
Input: Festival f
if GetAgentRole() ≠ AUDIENCE ∨ behavior.IsFighting() then
    //They do not care about the festival
    return;
if GetMoodType() = BORED then
    //Distress and resentment can cause boredom
    RemoveEventEffect(f);
    return;
eventExists ⇐ FALSE;
eventConfirmed ⇐ FALSE;
foreach g ∈ appraisal.Goals do
    if g.RelatedEvent = f then
        eventExists ⇐ TRUE;
        if ConseqForSelf(g) ∧ ProspectRelevant(g) ∧ Unconfirmed(g) ∧ Pleased(g) then
            if Within concert area ∧ ‖vel‖ < ε then
                //Agents already slow down if there are others in front
                g.Confirmed ⇐ CONFIRMED;
                eventConfirmed = TRUE;
                break;
if eventConfirmed then
    //Leading to joy
    appraisal.AddGoal(f, ConseqForSelf, ProspectIrrelevant, Pleased);
    //Standard about the singer
    appraisal.AddStandard(f, FocusingOnOther, Approving);
else
    if ¬eventExists then
        appraisal.AddGoal(f, ConseqForSelf, ProspectRelevant, Unconfirmed, Pleased);
```

Algorithm 5 describes the appraisal states of an agent from the audience in case there is a fight.

Algorithm 5: ComputeFightEffect: appraisal states of an agent in a fight

```
Input: Fight f
eventExists ⇐ FALSE;
//Agents witnessing a fight get distressed
if ¬IsFighting() ∧ GetAgentRole() ≠ PROVOCATEUR then
    foreach g ∈ appraisal.Goals do
        if g.RelatedEvent = f then
            dist = ||pos − f.pos||;
            eventExists ⇐ TRUE;
    if ¬eventExists ∧ dist < threshold then
        appraisal.AddGoal(f, ConseqForSelf, ProspectIrrelevant, Displeased);
if IsFighting() then
    AddDamage();
    opponent ⇐ GetOpponent();
    if IsWounded() ∨ opponent.IsWounded() then
        SetFighting(FALSE);
        opponent.SetFighting(FALSE);
```

It's always a provocateur who triggers a fight. In addition, the provocateur determines the start time and duration of the fight, taking control. Algorithm 6 shows the steps of fight for a provocateur. Algorithm 7 demonstrates the appraisal states for a provocateur.

Algorithm 6: PlanNextStep: steps of fight for a provocateur

```
if GetFighting() then
    if IsWounded then
        SetFighting(FALSE);
        opponent.SetFighting(FALSE);
        pos_attractor ⇐ pos_attractorInitial;
    else
        pos_attractor ⇐ opponent.GetPos();
else
    //Find someone to attack if not already fighting
    minDist ⇐ ∞;
    foreach Agent a ∈ GetVisibleAgents() do
        if a.GetAgentRole() ≠ PROVOCATEUR ∧ a.GetAgentRole() ≠
        SECURITY ∧ a.GetAgentRole() ≠ SINGER then
            dist ⇐ ||pos − a.pos||;
            if dist < minDist then
                opponent ⇐ a;
                minDist ⇐ dist;
    if minDist < catchDist then
        StartFighting(opponent);
    if minDist < ∞ then
        //Follow the victim to fight
        pos_attractor ⇐ opponent.GetPos();
```

Algorithm 7: StartFighting: appraisal states for a provocateur

```
Input: Opponent o
f ⇐ new Fight(o);
//Leading to gloating
appraisal.AddGoal(f, ConseqForOther, Undesirable, Pleased);
appraisal.AddStandard(f, FocusingOnOther, Disapproving);
SetFighting(TRUE);
//Opponent's appraisal status
//Leading to distress
o.appraisal.AddGoal(f, ConseqForSelf, ProspectIrrelevant, Displeased);
//Leading to resentment
o.appraisal.AddGoal(f, ConseqForOther, Desirable, Displeased);
o.appraisal.AddStandard(f, FocusingOnOther, Disapproving);
o.SetFighting(TRUE);
```

Figure 4.2 shows the state diagram of gesture updates according to moods for crowds in a festival.

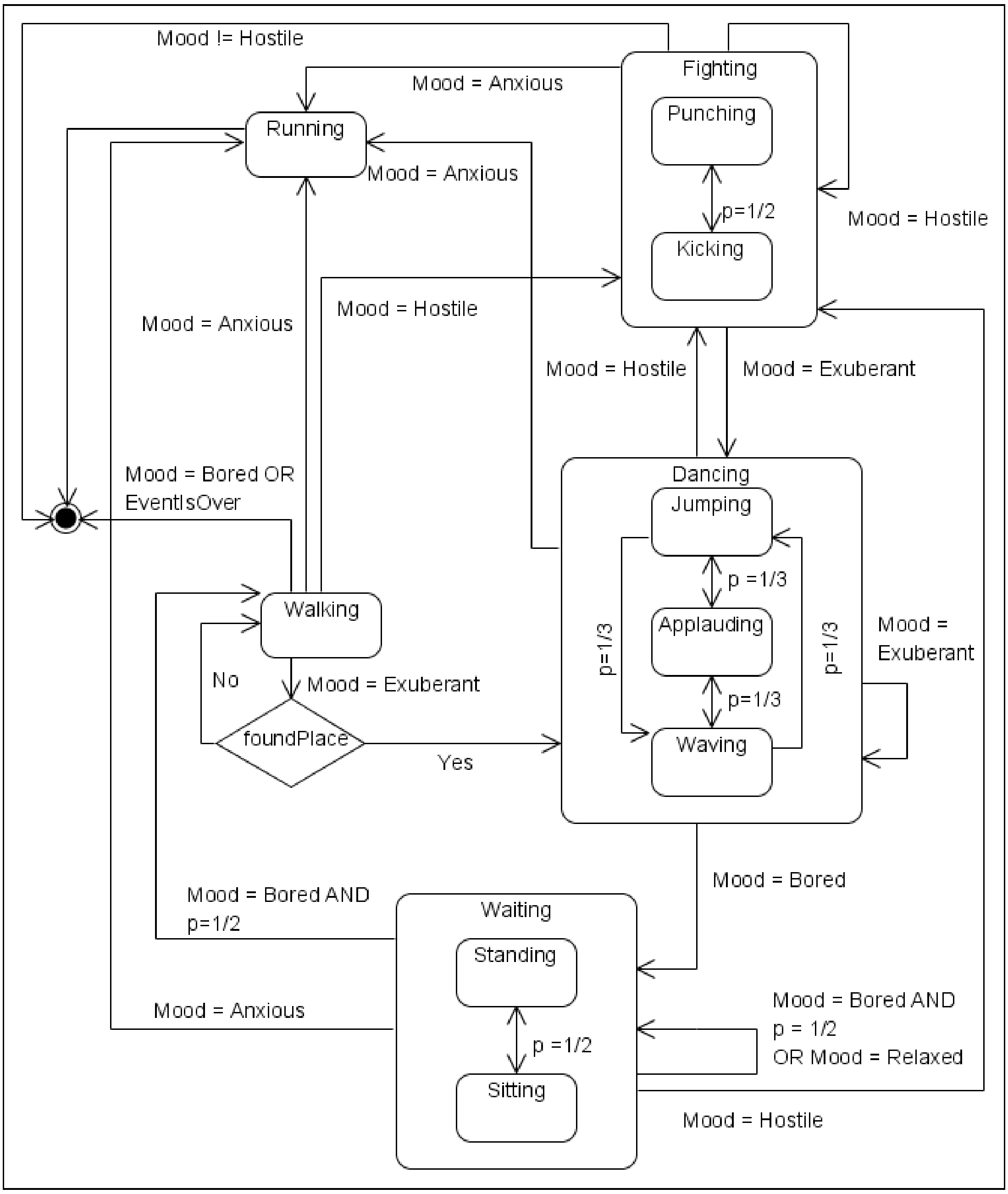


Figure 4.2: State diagram for gesture updates by mood in a festival

4.2.2 Protest

The protest scenario consists of mobs of angry agents marching down the streets, following a leader. The agent roles playing part in this scenario are protesters, their leaders and security officers. The following emotions are triggered in case of a protest:

Role: `Protester`
Goal: `March with peers in order to protest something`
  State: `Any`
    `Goals` → `Consequences for self` → `Prospect irrelevant` →
    `Displeased`→ `Distress`
Standard: `People subject to the protest`
  State: `Any`
    `Standards` → `Focusing on other` → `Disapproving` → `Reproach`
Standard: `Security`
  State: `When intervened by security`
    `Standards` → `Focusing on other` → `Disapproving` → `Reproach`
Compound Emotion: `Distress + Reproach = Anger`
Standard: `Self`
  State: `Any`
    `Standards` → `Focusing on self` → `Approving` → `Pride`
Standard: `Other protesters`
  State: `Any`
    `Standards` → `Focusing on other` → `Approving` → `Admiration`

Protesters have initial assessments about the protested situation and they have emerging standards about the security officials intervening. Algorithms 8 and 9 show the appraisal update for protesters.

Algorithm 8: InitProtest: initiating the protest

Input: Protest p
`//For the subjects of the protest`
appraisal.AddStandard(p, FocusingOnOther, Disapproving);
`//For other protesters`
appraisal.AddStandard(p, FocusingOnOther, Approving);
`//For themselves`
appraisal.AddStandard(p, FocusingOnSelf, Approving);
`//Leading to distress`
appraisal.AddGoal(p, ConseqForSelf, ProspectIrrelevant, Displeased);

Algorithm 9: PlanNextStep: appraisal update for protesters

Input: Protest p
foreach $s \in SecurityAgents$ **do**
  $dir \Leftarrow s.pos - pos$;
  `//Check if security and protester are facing each other`
  $\alpha \Leftarrow \arccos(-\frac{dir \bullet orientation}{\|dir\| * \|orientation\|})$;
  if $\alpha < \frac{\pi}{2} \wedge \|dir\| < threshold$ **then**
    `//Means agent got intervened by security`
    appraisal.AddStandard(p, FocusingOnOther, Disapproving);

`//Follow the leader`
$pos_{attractor} \Leftarrow leader.GetPos()$;

Figure 4.3 shows the state diagram of gesture updates for protesters according to moods. Please note that walking or standing states are concurrent with protesting or fighting. For instance, an agent can both applaud and walk or stand still at the same time. Therefore, we omitted these in the state diagram.

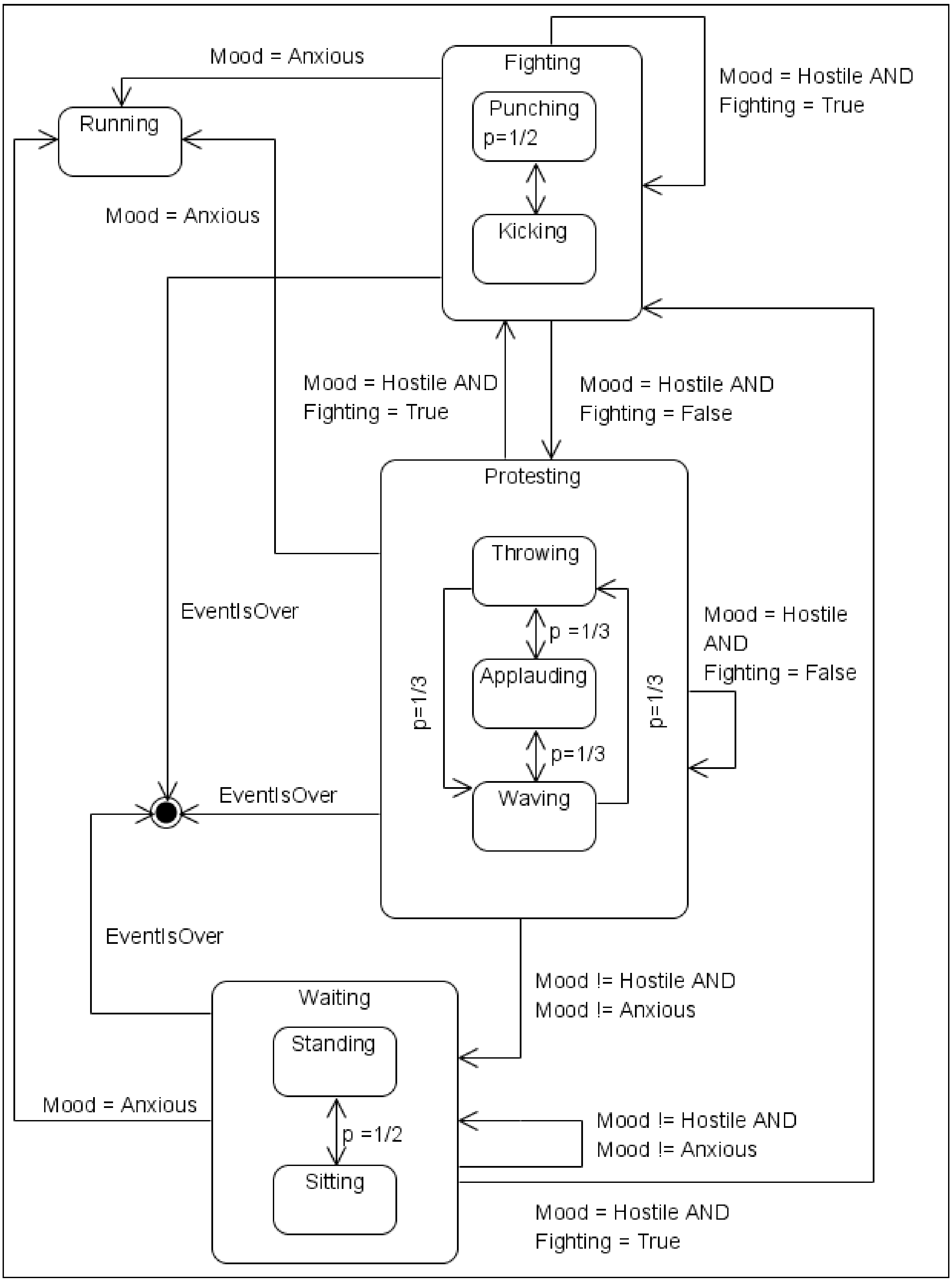


Figure 4.3: State diagram for gesture updates by mood in a protest

4.3 Escape Mobs

Escape mobs are simulated by creating an explosion scenario. The following emotions are triggered in case of an explosion:

Role: `Any`

Goal: `Run away from danger`

State: `Running`

`Goals` → `Consequences for self` → `Prospect relevant` → `Unconfirmed` → `Displeased` → `Fear`

State: `Managed to escape`

`Goals` → `Consequences for self` → `Prospect relevant` → `Disconfirmed` → `Displeased`→ `Relief`

State: `Caught by fire`

`Goals` → `Consequences for self` → `Prospect relevant` → `Confirmed` → `Displeased`→ `FearsConfirmed`

Algorithm 10 shows how state transitions are applied in case of an explosion. Agents get some damage depending on their distance to the center of explosion. Damage rules are applied according to the contagion equations given in Chapter 3. Getting infected means getting killed in the explosion. Of course, emotions have no meaning for a dead agent; however, we still apply the rules for confirmed fear, which is the last emotion that the agent experiences. Also, all the other events lose their meanings in case of a dangerous situation. Therefore, we remove all the events and their effects on the agents but explosion.

Algorithm 10: ComputeExplosionEffect: application of state transitions in an explosion

Input: Explosion e
$dist = \|pos - e.pos\|$;
if $dist >$ affectingDist **then**
  foreach $g \in appraisal.Goals$ **do**
    if $g.RelatedEvent = e \ \wedge \ Unconfirmed(g)$ **then**
      $g.Confirmed \Leftarrow$ DISCONFIRMED;
      break;
else
  `//Add damage negatively correlated with the distance to explosion`
  AddDamage(dist);
  eventExists $\Leftarrow$ FALSE;
  foreach $g \in appraisal.Goals$ **do**
    if $g.RelatedEvent = e \ \wedge \ UnConfirmed(g) \ \wedge$ *IsInfected()* **then**
      eventExists $\Leftarrow$ TRUE;
      $g.Confirmed \Leftarrow$ CONFIRMED;
  if $\neg$*eventExists* **then**
    `//Leading to fear`
    appraisal.AddGoal(e, ConseqForSelf, ProspectRelevant, Unconfirmed, Displeased);
    RemoveAllEventsButExplosion();

The physical computations of running away from the danger zone are given in *RepulsionFromEvents* procedure. Figure 4.4 shows the state diagram of gesture updates for escape mobs.

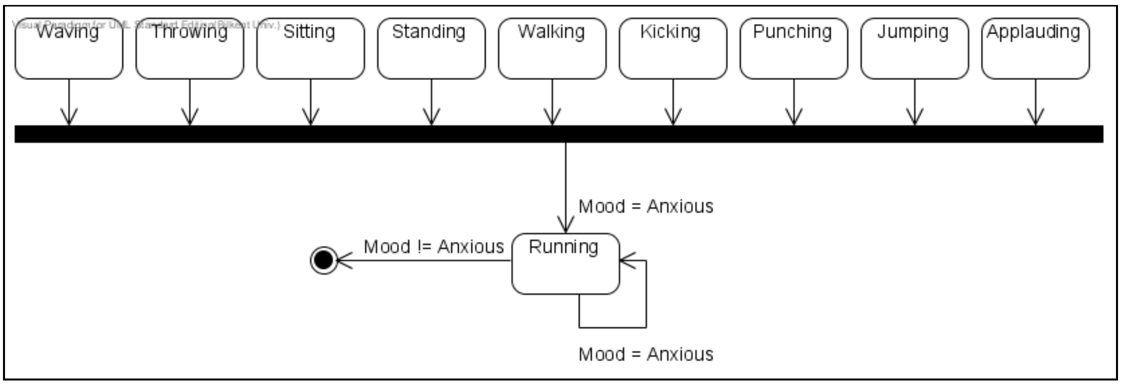


Figure 4.4: State diagram for gesture updates by mood in an explosion

4.4 Acquisitive Mobs

Acquisitive mobs are simulated in scenario that includes a sales event. Agents rush to a store to get an item for free. The following emotions are triggered in such a scenario:

Role: `Any`

Goal: `Get into the store`

  State: `Waiting`

    `Goals → Consequences for self → Prospect relevant → Unconfirmed → Pleased → Hope`

    `Goals → Consequences for self → Prospect irrelevant → Displeased → Distress`

  State: `All resources consumed`

    `Goals → Consequences for self → Prospect relevant → Disconfirmed → Pleased → Disappointment`

    `Goals → Consequences for other → Desirable for other → Displeased → Resentment`

    `Goals → Consequences for self → Prospect irrelevant → Displeased → Distress`

  State: `Managed to get some items`

    `Goals → Consequences for self → Prospect relevant → Confirmed → Pleased → Satisfaction`

Standard: `Others`

  State: `Too crowded, there is a stampede`

    `Standards → Focusing on other → Disapproving → Reproach`

Attitude: `Items in the store`

  State: `Any`

    `Attitudes → Liking → Love`

Compound Emotion: `Distress + Reproach = Anger`

Algorithm 11 shows how state transitions are applied in case of a sales event.

Algorithm 11: ComputeSalesEffect: application of state transitions in a sales event

```
Input: Sales s
dist = || pos − pos_item||;
eventExists ⇐ FALSE;
eventConfirmed ⇐ FALSE;
foreach g ∈ appraisal.Goals do
    if g.RelatedEvent = s then
        eventExists ⇐ TRUE;
        if ConseqForSelf(g) ∧ UnConfirmed(g) ∧ dist < ϵ ∧ GetItemCnt() > 0
        then
            s.DecreaseItemCnt();
            g.Confirmed ⇐ CONFIRMED;
            eventConfirmed ⇐ TRUE;
            //Update goals of other agents
            foreach Agent a ∈ OtherAgents do
                foreach g_o ∈ a.appraisal.Goals do
                    if g_o.RelatedEvent = s ∧ a.ConseqForSelf(g_o) then
                        //Leading to resentment
                        appraisal.AddGoal(s, ConseqForOther, Desirable, Displeased);
//Remove goals about others if an item is achieved
if eventConfirmed then
    appraisal.RemoveGoal(s, ConseqForOther);
if ¬eventExists then
    //Leading to hope
    appraisal.AddGoal(s, ConseqForSelf, ProspectRelevant, Pleased, Unconfirmed);
    appraisal.AddAttitude(s, Liking);
ComputeCrowdingEffect();
```

The method *ComputeCrowdingEffect* (Algorithm 12) updates the standards and goals of an agent in case the environment gets too crowded. Since crowding effect is considered an implicit event, when we add a goal, standard or attitude about the crowding effect, we do not need to specify the id of the event.

Algorithm 12: ComputeCrowdingEffect: update the standards and goals of an agent in case the environment gets too crowded

```
if GetDensityAhead() > threshold then
    goalExists ⇐ FALSE;
    foreach g ∈ appraisal.Goals do
        if GetEventType(g) = CROWDING then
            goalExists ⇐ TRUE;
            break;
    if ¬goalExists then
        appraisal.AddGoal(CROWDING,
        ConseqForSelf, ProspectIrrelevant, Displeased);
    standardExists ⇐ FALSE;
    foreach s ∈ appraisal.Standards do
        if GetEventType(s) = CROWDING then
            standardExists ⇐ TRUE;
            break;
    if ¬standardExists then
        appraisal.AddStandard(CROWDING,FocusingOnOther, Disapproving);
else
    //If not so dense, remove related goals and standards
    foreach g ∈ appraisal.Goals do
        if GetEventType(g) = CROWDING then
            appraisal.RemoveGoal(g);
    foreach s ∈ appraisal.Standards do
        if GetEventType(s) = CROWDING then
            appraisal.RemoveStandard(s);
```

Figure 4.5 shows the state diagram of gesture updates for audiences according to moods.

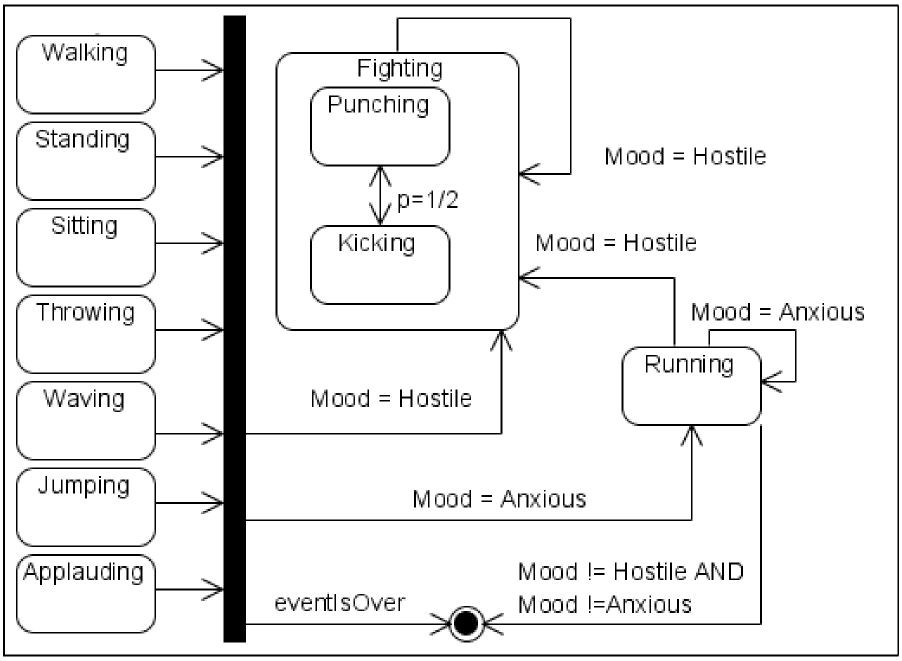


Figure 4.5: State diagram for gesture updates by mood in a sales event

4.5 Aggressive Mobs

The physical aspect of aggressive mobs is simulated by imitating predator-prey behavior. Here, attackers act like predators and victims act like preys [69]. Let V be the set of victims and A be the set of attackers. Given an attacker $att \in A$ with a position of pos_a and a victim $vic \in V$ with a position of pos_{vic}, the avoidance force f_{av} of victim vic from the attacker a is computed as follows:

$$f_{av} = c_{av} \frac{pos_{vic} - pos_{att}}{1 + \exp(\omega(\|pos_{vic} - pos_{att}\| - r))}, \tag{4.3}$$

where r is the visibility radius of the victim. The model ensures that victims run away from an attacker when the attacker is visible to them. The constant ω determines the degree of fall-of for the avoidance force.

Attacker behavior is handled differently. Attackers (or predators) do not tend to work in groups; their only tendency is to catch victims (or preys). The governing equations for the control of the movement of an attacker a are:

$$\begin{aligned}
targetPos_{att} &= argmin(pos_{att} - pos_{vic}),\ vic = 1, \ldots, \|V\| \\
desV_{att} &= \frac{targetPos_{att} - pos_{att}}{\|targetPos_{att} - pos_{att}\|} \\
f_{att} &= c_{att} \frac{desV_{att} - v_{att}}{\|desV_{att} - v_{att}\|}
\end{aligned} \tag{4.4}$$

where $targetPos_{att}$ is the target position, which is the closest victim visible to the attacker, $desV_{att}$ is the desired velocity and f_{att} is the attack force.

Damage conforms to the contagion rules. The victim is in one of two states: susceptible or infected. Getting infected means getting caught and killed. When the victim is killed, it falls down and becomes an obstacle for other agents.

Role: `Attacker`

Goal: `Catch a victim`

State: `Chasing`

`Goals` → `Consequences for self` → `Prospect relevant` → `Unconfirmed` → `Pleased` → `Hope`

State: `Caught someone`

`Goals` → `Consequences for self` → `Prospect relevant` → `Confirmed` → `Pleased` → `Satisfaction`

`Goals` → `Consequences for other` → `Undesirable for other` → `Pleased` → `Gloating`

State: `Missed all`

`Goals` → `Consequences for self` → `Prospect relevant` → `Disconfirmed` → `Pleased` → `Disappointment`

`Goals` → `Consequences for other` → `Undesirable for other` → `Displeased` → `Resentment`

Standard: `Victims`

State: `Any`

`Standards` → `Focusing on other` → `Disapproving` → `Reproach`

<u>Role:</u> <u>`Victim`</u>

Goal: `Escape`

  State: `Running`

    `Goals` → `Consequences for self` → `Prospect relevant` →
    `Unconfirmed` → `Displeased` → `Fear`

  State: `Got caught`

    `Goals` → `Consequences for self` → `Prospect relevant` →
    `Confirmed` → `Displeased` → `FearsConfirmed`

    `Goals` → `Consequences for other` → `Desirable for other` →
    `Displeased` → `Resentment`

  State: `Managed to escape`

    `Goals` → `Consequences for self` → `Prospect relevant` →
    `Disconfirmed` → `Displeased` → `Relief`

    `Goals` → `Consequences for other` → `Undesirable for other` →
    `Pleased` → `Gloating`

Standard: `Attackers`

  State: `Any`

    `Standards` → `Focusing on other` → `Disapproving` → `Reproach`

Algorithms 13 and 14 present how an attacker constructs his/her attacking plan.

Algorithm 13: InitAttack: initiating an attacker's attacking plan

$a \Leftarrow$ *new Attack()*;
`//Hope to catch a victim`
appraisal.AddGoal(a, ConseqForSelf, ProspectRelevant, Unconfirmed, Pleased);
appraisal.AddStandard(a, FocusingOnOther, Disapproving);

Algorithm 14: PlanNextStepAttack: planning the next steps of an attacker's attacking plan

Input: Attack a
minDist $\Leftarrow \infty$;
foreach $v \in$ *GetVisibleVictims()* **do**
  $dist = \|v.pos - pos\|$;
  if $dist < minDist$ **then**
    $victim \Leftarrow v$;
    $minDist \Leftarrow dist$;

`//If caught a victim`
if $minDist <$ catchDist **then**
  foreach $g \in appraisal.Goals$ **do**
    if $g.RelatedEvent = a \ \wedge$ *Unconfirmed(a)* **then**
      `//Leading to satisfaction`
      $g.Confirmed \Leftarrow$ CONFIRMED;
      `//Leading to gloating`
      appraisal.AddGoal(a, ConseqForOther, Undesirable, Pleased);

if $minDist = \infty$ **then**
  return;
$\overleftarrow{dir} \Leftarrow v.pos - pos$;
$vel_{desired} \Leftarrow \frac{\overleftarrow{dir}}{\|\overleftarrow{dir}\|} maxSpeed$;

Algorithms 15 and 16 show the steps of the victim's escape plan.

Algorithm 15: InitEscape: initializing the victim's escape plan

Input: Attack a
`//Leading to fear`
appraisal.AddGoal(a, ConseqForSelf, ProspectRelevant, Unconfirmed, Displeased);
appraisal.AddStandard(a, FocusingOnOther, Disapproving);

Algorithm 16: PlanNextStepEscape: planning the next step for the victim's escape plan

```
Input: Attack a
//Run away from the center of visible attackers
centerAtt ⇐ 0;
countAtt ⇐ 0;
foreach a ∈ GetVisibleAttackers() do
    centerAtt = centerAtt + a.pos;
    countAtt = countAtt + 1;
centerAtt = centerAtt / countAtt;
//If caught
if ‖centerAtt − pos‖ < catchDist then
    AddDamage();
    if IsInfected() then
        GetKilled();
        foreach g ∈ appraisal.Goals do
            if g.RelatedEvent = a ∧ Unconfirmed(a) then
                //Leading to fearsConfirmed
                g.Confirmed ⇐ CONFIRMED;
        return;
    else
        dir_avoid = pos − centerAtt;
shelter ⇐ FindClosestShelter();
if ‖centerAtt − pos‖ > ‖shelter.pos − pos‖ then
    //Go to the closest shelter
    f_avoid = shelter.pos − pos;
else
    //Avoid attackers
    dir_avoid = pos − centerAtt;
```

Figure 4.6 shows the state diagram of gesture updates for attackers according to moods. Figure 4.7 shows the state diagram of gesture updates for victims according to moods.

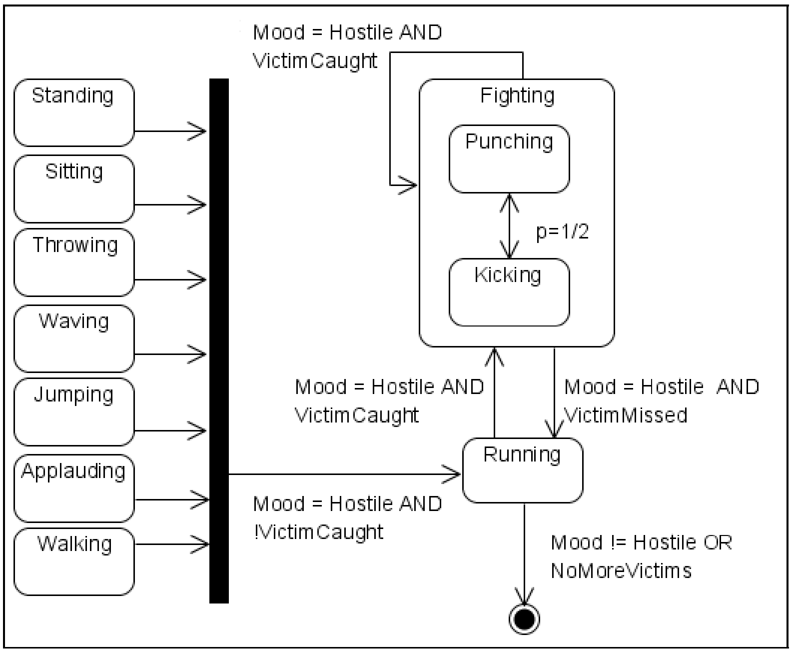


Figure 4.6: State diagram for gesture updates by mood by an attacker

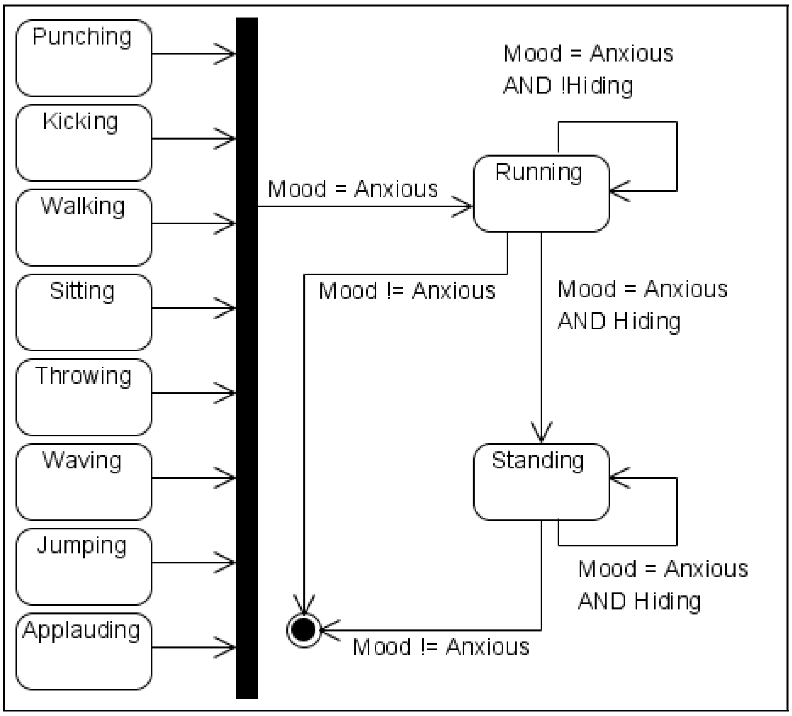


Figure 4.7: State diagram for gesture updates by mood by a victim

Chapter 5

Experiments and Results

5.1 User Studies on Personality

We analyze the overall emergent crowd behaviors considering personality-to-behavior mapping. We validate our hypotheses by user studies that assess the perception of the traits in the animations illustrating such behaviors. We created several animations to see how global crowd behavior is affected by modifying the personality parameters of subgroups.

5.1.1 Design of the Experiment

We created 15 videos presenting the emergent behaviors of people in various scenarios where the crowds' behavior is driven by the settings assigned through the OCEAN model. The scenarios range from evacuation drills to cocktail parties or museum galleries.

The mapping from HiDAC parameters to OCEAN factors is done through trait-descriptive adjectives. We find the correspondence between our mapping and the users' perception of these trait terms in the videos in order to validate our system. 70 subjects (21 female, 49 male, ages 18-30) participated in the experiment. We showed the videos to the participants through a projected display and asked them to fill out a questionnaire consisting of 123 questions– about 8 questions per video. The videos were shown one by one; after each video, participants were given some time to answer the questions related to the video. The participants did not have any prior knowledge about the experiment. Questions assess how much a person agrees with statements such as “I think the people in this video are kind.” or “I think the people with green suits are calm.” We have used questions containing the adjectives that describe each of the OCEAN factors instead of asking directly about the OCEAN factors, since we consider that the general public, not being familiar with the OCEAN model could have difficulties answering questions such as “Do the people exhibit openness?” Although the participants are proficient in English, in order to prevent any misconceptions, definitions of the adjectives were attached to the questionnaires. Definitions were taken from the Merriam-Webster dictionary. The answers were selected from a scale between 0 and 10, increasing by 1, where 0 = totally disagree, 5 = neither agree nor disagree, 10 = totally agree. We omitted the antonyms from the list of adjectives for the sake of conciseness. Thus, the remaining adjectives were: *assertive*,*calm*, *changeable*, *contrary*, *cooperative*, *curious*, *distant*, *energetic*, *harsh*, *ignorant*, *kind*, *orderly*, *patient*, *predictable*, *rude*, *shy*, *social*, *stubborn*, and *tolerant*.

5.1.2 Sample Scenarios

The simulated scenarios help us observe how the suggested parameters affect the global behavior of a crowd. In the implemented settings, novel, emergent formations are realized and behavior timings are also affected. We explain a selection of scenarios that have been shown to the participants in our experiments.

A sample scenario testing the impact of openness takes place in a museum setting as one of the key factors determining openness is the belief in the importance of art. A screenshot from the sample animation can be seen in Figure 5.1. *Curiosity* and *ignorance* are the tested adjectives for this setting. There are three groups of people, with openness values 0, 0.5 and 1. Here, the number of tasks that each agent must perform is mapped to openness, where a task means looking at a painting. The least open agents (with blue hair) leave the museum first, followed by the agents with openness values of 0.5 (with black hair). The most open agents (with red hair) stay the longest. Participants are asked how they perceive each of these groups.

Figure 5.1: Openness tested in a museum. The most open people (red-heads) stay the longest, whereas the least open people (blue-heads) leave the earliest.

Another one of our videos assesses how extroverts and introverts are perceived according to their distribution around a point of attraction. Figure 5.2 shows a screenshot from our test video where the agents in blue suits are extroverted with $\mu = 0.9$ and $\sigma = 0.1$ and the agents in grey suits are introverted with $\mu = 0.1$ and $\sigma = 0.1$. The ratio of introverts to extroverts in a society is found to be 25%, according to which we assigned the initial number of agents [68]. At the end of the animation, introverts are left out of the ring structure around the object of attraction. As extroverts are faster, they approach the attraction point in a shorter time. In addition, when there are other agents blocking their way, they tend to push them to reach their goal. The figure also shows the difference between the personal spaces of individuals with introverted and extroverted personality. Thus, being *social*, *distant*, *assertive*, *energetic*, and *shy* is questioned for this animation.

In order to test whether the personalities of people creating congestion are distinguished, we showed the participants two videos of same duration and asked them to compare the characteristics of the agents in each video. Each video consists of two groups of people moving through each other. The first video shows people with high agreeableness and conscientiousness values ($\mu = 0.9$ and $\sigma = 0.1$ for both traits), whereas the second video displays people with low agreeableness and conscientiousness values ($\mu = 0.1$ and $\sigma = 0.1$ for both traits). In the first video, groups manage to cross each other while in the second video congestion occurs after a fixed period of time. Such behaviors emerge as agreeable and conscientious individuals are more patient; they do not push each other and are always predictable as they prefer the right side to move on. Figure 5.3 shows how congestion occurs due to low conscientiousness and agreeableness values. People are stuck at the center, and they refuse to let other people move, thus they are also *stubborn*, *negative*, and not *cooperative*.

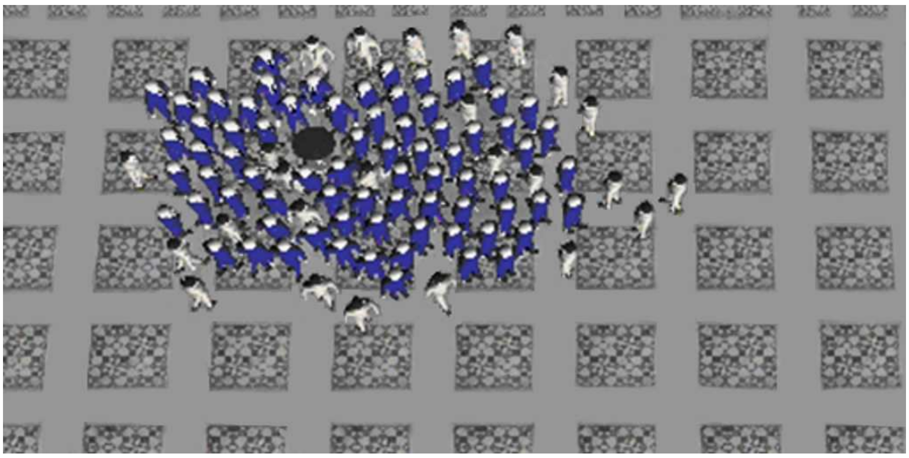


Figure 5.2: Ring formation where extroverts (blue suits) are inside and introverts are outside

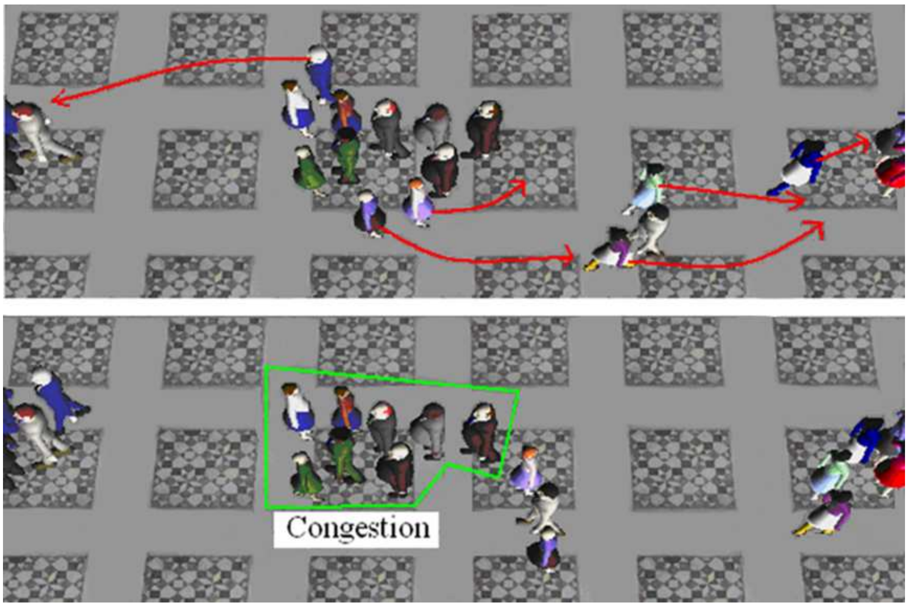


Figure 5.3: People with low conscientiousness and agreeableness value cause congestion.

Figure 5.4 shows a screenshot from the animation demonstrating the effect of neuroticism, non-conscientiousness and disagreeableness on panic behavior. A total of 13 agents are simulated. Five of the agents have neuroticism values of $\mu = 0.9$ and $\sigma = 0.1$, conscientiousness values of $\mu = 0.1$ and $\sigma = 0.1$ and agreeableness values of $\mu = 0.1$ and $\sigma = 0.1$. The remaining agents, which are stable, have neuroticism values of $\mu = 0.1$ and $\sigma = 0.1$, conscientiousness values of $\mu = 0.9$ and $\sigma = 0.1$ and agreeableness values of $\mu = 0.9$ and $\sigma = 0.1$. The agents in green suits are neurotic, less conscientious, and disagreeable. It can be seen in the figure that these agents tend to panic more, push other agents, force their way through the crowd, and rush to the door. These agents are not *predictable*, *cooperative*, *patient*, or *calm* but they are *rude*, *changeable*, *negative*, and *stubborn*.

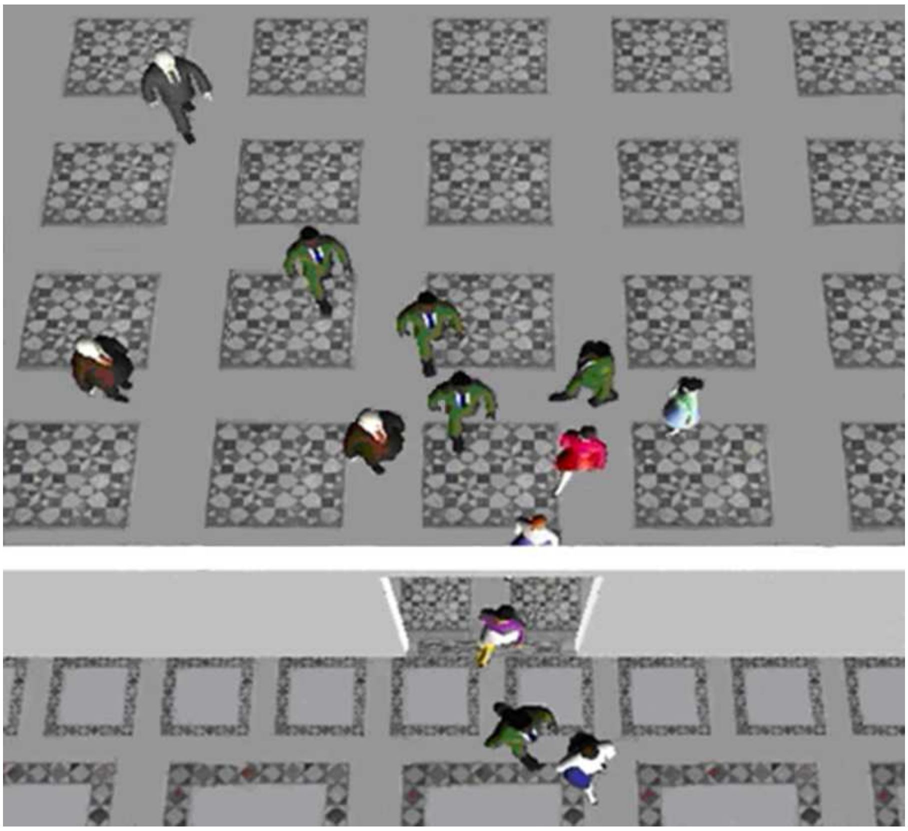


Figure 5.4: Neurotic, non-conscientious and disagreeable agents (in green suits) show panic behavior.

5.1.3 Analysis

After collecting the participants' answers for all the videos, we first organized the data for the adjectives. Each adjective is classified by its question number, the actual simulation parameter and the participants' answers for the corresponding question. We calculated the Pearson correlation (r) between the simulation parameters and the average of the subjects' answers for each question. For instance the adjective *assertive* is asked 8 times, which indicates a sample size of 8. Thus, the correlation coefficient between the actual parameters and the means of the participants' answers is calculated between these 16 values, 8 for each group.

Furthermore, we grouped the relevant adjectives for each OCEAN factor in order to assess the perception of personality traits, which is the actual purpose of our experiment. The evaluation process is similar to the evaluation of adjectives; this time considering the questions for all the adjectives corresponding to an OCEAN factor. For instance, as openness is related to *curiosity* and *ignorance*, the answers for both of these adjectives is taken into account. Again, we averaged the subjects' answers for each question; then, we computed the correlation with the actual parameters and the mean throughout all the questions asking for *curious* and *ignorant*.

In order to estimate the probability of having obtained the correlation coefficients by chance, we computed the significance of the correlation coefficients. Significance is taken as $1 - p$, where p is the two-tailed probability that is calculated considering the sample size and the correlation value. Higher correlation and significance values suggest more accurate user perception.

5.1.4 Results and Discussion

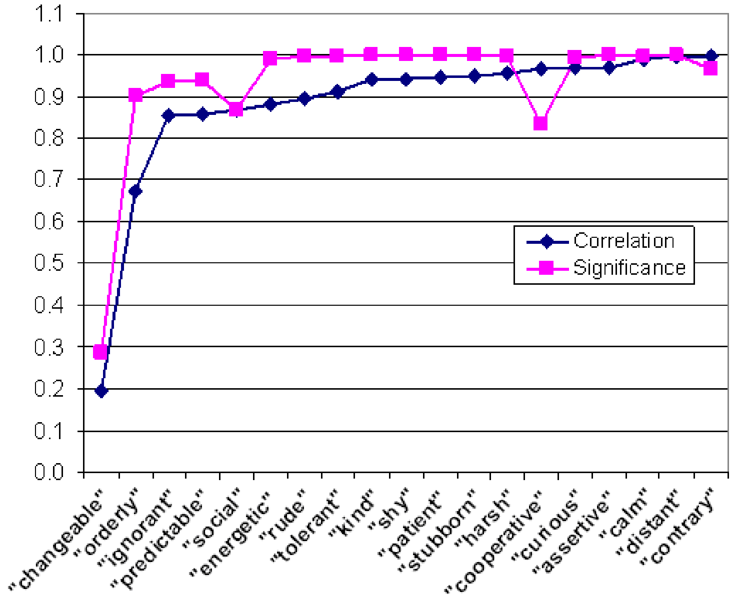
The correlation coefficients and significance values for the adjectives are depicted in Figure 5.5 along with the data table showing the exact results. Correlation values are sorted in ascending order. The pink data points indicate the significance of the correlation coefficients. As can be seen from the data table, significance is low (< 0.95) for the adjectives *changeable*, *orderly*, *ignorant*, *predictable*, *social* and *cooperative*. Low significance is caused by low correlation values for *changeable* and *orderly*. However, although the correlation coefficients are found to be high for *predictable*, *ignorant*, *social* and *cooperative*, low significance can be explained due to small sample size.

From the participants' comments, we figured out that the term *changeable* is especially confusing. In order to understand the reason, we can consider the aforementioned setting where two groups of agents cross each other. Non-conscientious agents are identified as *rude*, however; they are perceived as persistent in their rudeness, causing the participants to mark lower values for the question asking changeability. The same problem holds for *predictable* as well. One of the participants' comments suggest that if a person is in a rush, you can predict that person to push others. However, *predictable* has higher correlation despite these comments and although it implies an opposite meaning to *changeable*. This could be due to the relatively low significance for *predictable*. Non-conscientious agents that cause congestion are perceived as less *predictable*, which indicates that changing right preference and rude behavior decreases the perceived predictability.

Orderly is another weakly correlated adjective. Analyzing the results for each video separately, we found out that agents in evacuation drill scenarios are found to be orderly, although they show panic behavior. In these videos, even if the agents push each other and move fast, still some kind of order can be observed. This is due to the smooth flow of the crowd during building evacuation. The crowd shows collective synchrony, where individuality is lost. Although individuals are impatient and rude, the overall crowd behavior appears *orderly*. We assigned the same goal to the entire crowd in evacuation simulations, because our aim was to observe disorganization locally. For instance, *disorderly* agents look in a rush; they push other agents and they do not have solid preferences for direction choosing when crossing an agent in an evacuation scenario. Nevertheless, they still move to the same goal, which is the exit of the building. The crowd would appear more *disorderly* if everyone were running in different directions and changing directions for no apparent reason. Participants' answers suggest that they do not recognize orderliness where the goal is the same for the whole crowd. On the other hand, in another scenario, which shows the queuing behavior of a crowd in front of a water dispenser, participants can easily distinguish *orderly* versus *disorderly* individuals. Orderly agents wait at the end of the queue, whereas *disorderly* agents rush to the front. In this setting, although the main goal is the same for all the agents (drinking water), there are two distinguishable groups who act differently.

Figure 5.6 shows the correlation coefficients and their significance for the OCEAN parameters. These values are computed by taking into account all the relevant adjectives for each OCEAN factor. The correlations are sorted in ascending order. As can be seen from the figure, the significance of all the coefficients is high, with a probability of less than 0.5% of being by chance ($p < 0.005$). Significance is high because all the adjectives describing a personality factor are taken into account, achieving sufficiently large sample size.

Correlation coefficient for conscientiousness is comparatively low among all personality factors, showing that only about 44% of the traits are perceived correctly ($r^2 \approx 0.44$). In order to understand the underlying reason, we should consider the relevant adjectives, which are *orderly*, *predictable*, *rude* and *changeable*. Low correlation values for *orderly* and *changeable* reduce the overall correlation. If we consider only *rude* and *predictable* for conscientiousness, correlation increases by 18.6%. Thus, the results suggest that, people can observe the politeness aspect of personality in short-term crowd behavior settings more easily than the organizational aspects. This also explains why the perception of agreeableness is highly correlated with the actual parameters. Figure 5.6 also shows that neuroticism is perceived the best. In this study, we have only considered the calmness aspect of neuroticism, which is tested in emergency settings and building evacuation scenarios.

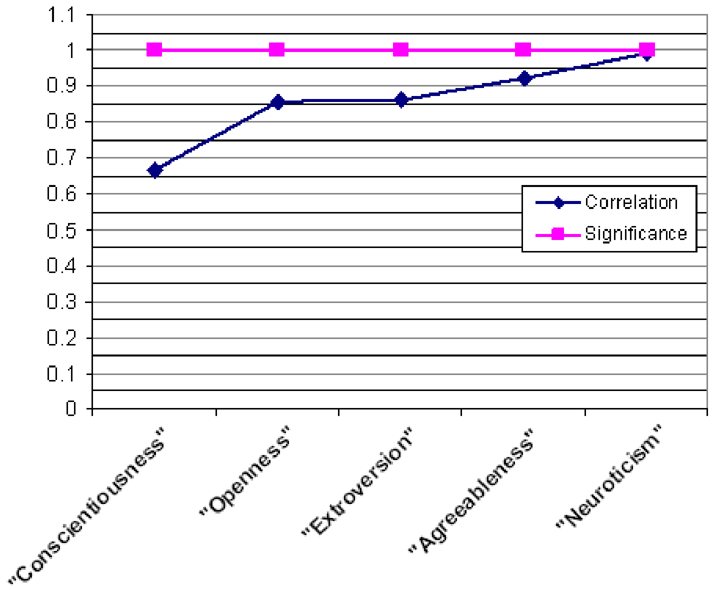


(a)

Adjective	Correlation	Significance
"changeable"	0.199	0.288
"orderly"	0.674	0.903
"ignorant"	0.853	0.936
"predictable"	0.870	0.938
"social"	0.872	0.869
"energetic"	0.882	0.992
"rude"	0.897	0.997
"tolerant"	0.912	0.998
"kind"	0.943	1.000
"shy"	0.945	1.000
"patient"	0.948	1.000
"stubborn"	0.950	1.000
"harsh"	0.956	0.997
"cooperative"	0.967	0.834
"curious"	0.971	0.994
"assertive"	0.971	1.000
"calm"	0.988	0.999
"distant"	0.998	1.000
"contrary"	0.999	0.969

(b)

Figure 5.5: (a) The graph depicts the correlation coefficients between actual parameters and subjects' answers for the descriptive adjectives (blue); significance values for the corresponding correlation coefficients (pink). (b) Data table showing the correlation coefficients and significance values for descriptive adjectives.



(a)

OCEAN	Correlation	Significance
"Conscientiousness"	0.665	1.000
"Openness"	0.859	0.999
"Extroversion"	0.860	1.000
"Agreeableness"	0.922	1.000
"Neuroticism"	0.990	0.999

(b)

Figure 5.6: (a) The graph depicts the correlation coefficients between actual parameters and subjects' answers for the OCEAN factors (blue); two-tailed probability values for the corresponding correlation coefficients (pink). (b) Data table showing the correlation coefficients and the significance values for the OCEAN factors.

5.2 Runtime Performance

The simulations are run on a personal computer (Intel Core Duo Processor E8400, 3.00GHz) with 3.24GB of RAM. The graphics card is ATI Radeon HD 3800 with 512 MB memory size. We use Cal3D Character Animation Library for rendering and animating the 3D human characters. The average frame rates for the simulation of crowds of different sizes is given in Figure 5.7

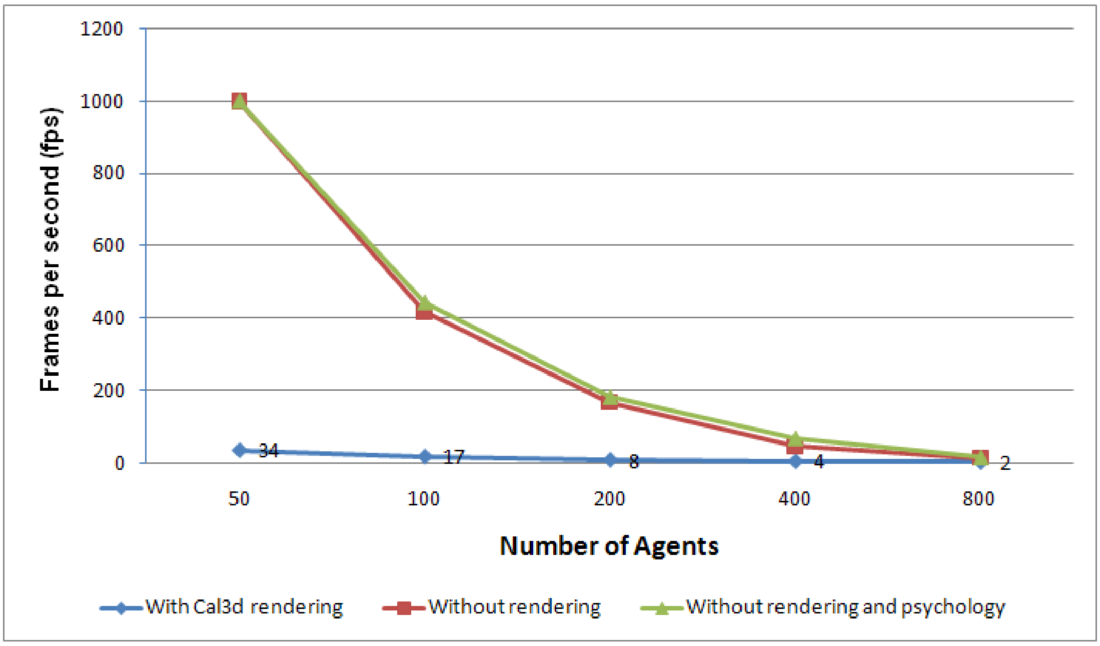


Figure 5.7: Frames rates (frames per second) for different sizes of crowds

We found similar frame rates for different scenarios. Therefore, we give the average time performance for all types of events. The results indicate that Cal3D rendering is the bottleneck of simulations. Even with 50 agents, time performance is below interactive rates. When rendering cost is excluded, we achieve real time simulation results with 200 agents and near-interactive frame rates with 400 agents. The results indicate that the psychological component does not bring much overhead to the actual HiDAC implementation.

5.3 Visual Results for Different Events

In this section, we present still frames from the simulations performed using our system. Figure 5.8 shows an explosion and a close-up view of a scared agent

Figure 5.8: Explosion scenario

running. Figure 5.9 shows a street concert with 400 attenders. Figure 5.10 shows a sales event with 200 people rushing into a store and their view inside the store. Figure 5.11 presents a protest scenario with 500 protesters and 60 security officers standing side-by-side, watching them.

(a)

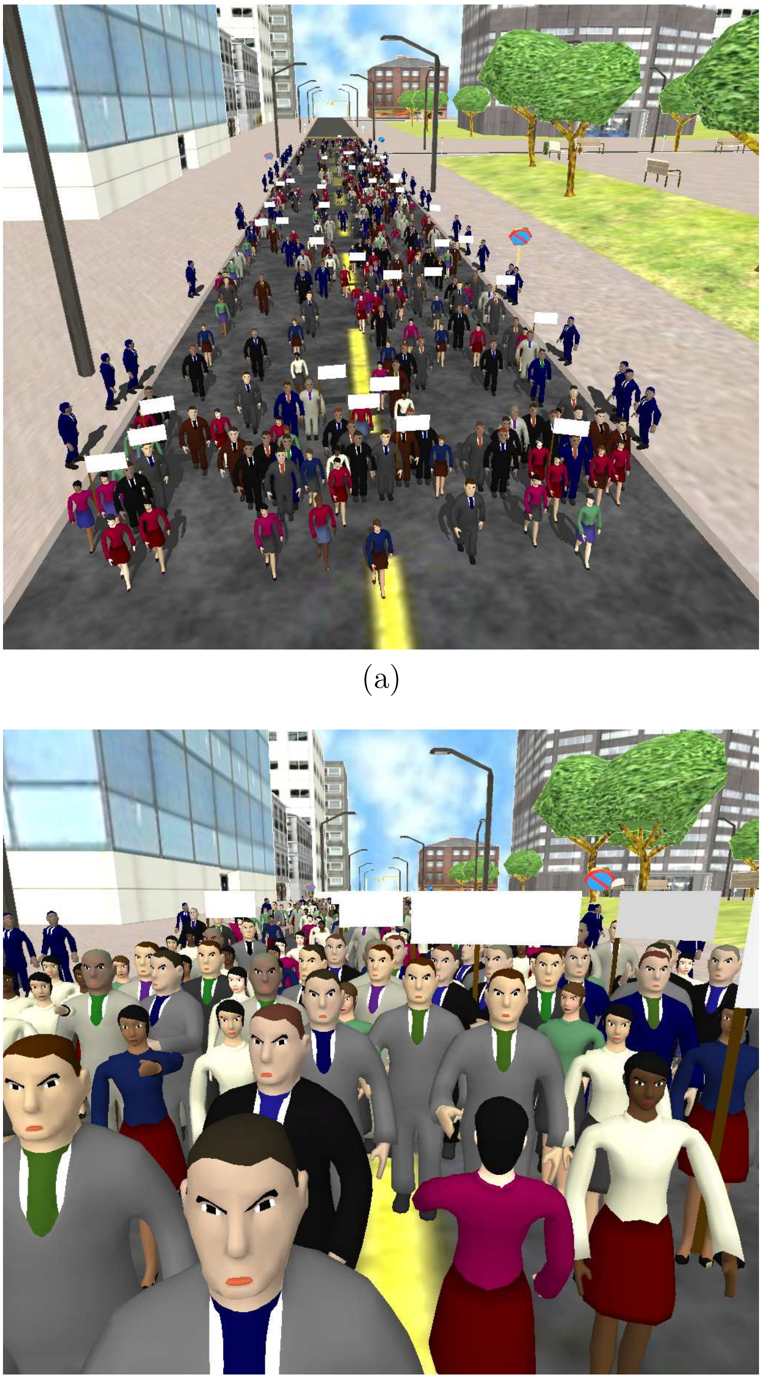
(b)

Figure 5.9: Festival scenario with (a) distant and (b) close-up views

(a)

(b)

Figure 5.10: Sales scenario (a) outside (b) inside a store



(a)

(b)

Figure 5.11: Protest scenario with (a) distant and (b) close-up views

Chapter 6

Conclusion

We propose a crowd simulation system that incorporates a complex psychological component into the agents. So far, autonomous agents research has focused on enhancing the believability of individual agents. In order to create a believable virtual human, different components comprising a real human must be considered. Intelligence by itself, for example, is not enough to represent the complexity of a human's interaction with the environment. Especially, conversational agents show human-like behavior by expressing their emotions. We integrated these facilities to a crowd simulation system. In our case, since there is a large number of virtual humans interacting with each other, psychological features of these humans become more significant. Furthermore, runtime results indicate that increasing the psychological complexity of agents does not bring much overhead to the simulation performance, which is promising for our purposes.

The psychological module is composed of three components: personality, mood and emotion. Personality is intrinsic; therefore, it is up to the user to determine which agents will have which personality traits. In that sense, we use the OCEAN personality model, which is well a respected and complete model to simulate personality traits [116]. Emotions and moods are then computed based on personality and how the agent perceives external events. We use the OCC model of emotions, which states that emotions are based on cognitive appraisal of events [89]. As for the moods, we use the PAD (Pleasure, Arousal, Dominance) model, which serves a connection between personality and emotions [78].

Crowd behavior has always drawn the attention of social psychologists. The reasons underlying why some crowds act temperamentally, losing sensibility, acting aggressively or panicking are still not fully understood. Theoreticians attempt to explain such phenomena by classifying crowds and developing theories about mass behavior. We utilize some of these theories to set a foundation for our system. In doing so, we incorporate predisposition theories with contagion theories, exploiting the most beneficial aspects of both sides for the sake of our design.

We design and simulate various scenarios, each corresponding to a different crowd type. More specifically, we are interested in mob behavior, and how regular crowds, i.e. audiences, turn into mobs. However, it is not the individual scenarios that is important here, but the functionality that our system provides. For instance, another programmer might have designed the scenarios in a different way. It is only a matter of defining your own rules for different situations. As a future work, we plan to enable the integration of different scenarios as plug-in programs.

Our future plans include creating a setting, in which an actual human user interacts with the system by being a part of the crowd through virtual reality equipment. We already have the functionality to include the user into the simulation and see the simulations through first person view from the screen. However, we plan to increase the sense of presence through head-mounted displays and motion capture equipment and validate our system in this way.

Bibliography

[1] V. Akman. *Unobstructed shortest paths in polyhedral environments*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[2] J. Allbeck and N. Badler. Toward representing agent behaviors modified by personality and emotion. In *Proceedings of Embodied Conversational Agents at AAMAS'02*, Bologna, Italy, July 2002.

[3] G. Allport. *Handbook of Social Psychology, G. Lindzey (ed.)*, chapter The Historical Background of Modern Social Psychology, pages 3–56. Addison-Wesley, 1954.

[4] Y. C. Alon Lerner and D. Cohen-Or. Efficient cells-and-portals partitioning. *Computer Animation and Virtual Worlds*, 17:21–40, 2006.

[5] M. Anderson, E. McDaniel, and S. Chenney. Constrained animation of flocks. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 286–297, 2003.

[6] D. Arellano, J. Varona, and F. J. Perales. Generation and visualization of emotional states. *Computer Animation and Virtual Worlds*, 19:259–270, 2008.

[7] O. Arikan, S. Chenney, and D. A. Forsyth. Efficient multi-agent path planning. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, pages 151–162, 2001.

[8] K. Ashida, S. Lee, J. Allbeck, H. Sun, N. Badler, and D. Metaxas. Pedestrians: Creating agent behaviors through statistical analysis of observation

data. In *Proceedings of IEEE Conference on Computer Animation*, pages 84–92, 2001.

[9] G. Ball and J. Breese. Emotion and personality in a conversational character. In *Proceedings of the Workshop on Embodied Conversational Characters*, page 8384 and 119121, 1998.

[10] O. Bayazit, J. Lien, and N. Amato. Better group behaviors in complex environments with global roadmaps. In *Proceedings of the Eighth International Conference on Artificial Life (Alife)*, pages 362–370, 2002.

[11] O. Bayazit, J. Lien, and N. Amato. Better group behaviors using rule-based roadmaps. In *Proceedings of International Workshop on Algorithmic Foundations of Robotics (WAFR)*, Nice, France, 2002.

[12] C. Becker, S. Kopp, and I. Wachsmuth. Simulating the emotion dynamics of a multimodal conversational agent. *Lecture Notes in Artificial Intelligence*, 3068:54–165, 2004.

[13] C. Becker-Asano and I. Wachsmuth. Affect simulation with primary and secondary emotions. *Lecture Notes in Artificial Intelligence*, 5208:15–28, 2008.

[14] R. A. Berk. *Collective Behavior*. Dubuque, Iowa: W.C. Brown, 1974.

[15] V. Blue and J. Adler. Cellular automata model of emergent collective bi-directional pedestrian dynamics. In *Proceedings of Artificial Life VII*, pages 437–445, 2000.

[16] B. M. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. Johnson, and B. Tomlinson. Integrated learning for interactive synthetic characters. *ACM Computer Graphics (Proceedings of SIGGRAPH'02)*, pages 417–426, 2002.

[17] B. M. Blumberg and T. A. Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. *ACM Computer Graphics (Proceedings of SIGGRAPH'95)*, pages 47–54, 1995.

[18] H. Blumer. *Principles of Sociology, A.M. Lee (ed.)*, chapter Collective Behavior, pages 67–121. Barnes & Noble, New York, 1951.

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

[20] G. H. Bower and P. R. Cohen. *Affect and Cognition, M. Clark and S. Fiske (eds.)*, chapter Emotional Influences in Memory and Thinking: Data and Theory, pages 291–331. Lawrence Erlbaum Associated Publishers, 1982.

[21] W. Breitfuss, H. Prendinger, and M. Ishizuka. Automatic generation of conversational behavior for multiple embodied virtual characters: The rules and models behind our system. *Lecture Notes in Artificial Intelligence*, 5208:472–473, 2008.

[22] D. C. Brogan and J. K. Hodgins. Group behaviors for systems with significant dynamics. *Autonomous Robots*, 4:137–153, 1997.

[23] D. C. Brogan, R. A. Metoyer, and J. K. Hodgins. Dynamically simulated characters in virtual environments. *IEEE Comput. Graph. Appl.*, 18(5):58–69, 1998.

[24] R. W. Brown. *The Handbook of Social Psychology, G. Lindzey (ed.)*, chapter Mass Phenomena, pages 833–876. Addison-Wesley, Cambridge, MA, 1954.

[25] R. Burke, D. Isla, M. Downie, Y. Ivanov, and B. Blumberg. Creature smarts: The art and architecture of a virtual brain. In *Proceedings of the Computer Game Developers Conference*, pages 147–166, 2001.

[26] S. Chenney. Flow tiles. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 233–242, 2004.

[27] R. M. Christopher Peters, Cathy Ennis and C. OSullivan. Crowds in context: Evaluating the perceptual plausibility of pedestrian orientations. In *Proceedings of Eurographics (Short Papers)*, 2008.

[28] T. Conde, W. Tambellini, and D. Thalmann. Behavioral animation for autonomous virtual agents helped by reinforcement learning. In *Lecture Notes in Computer Science (Proceedings of Intelligent Virtual Agents)*, volume 2792, pages 175–180, Berlin, Heidelberg, 2003.

[29] T. Conde and D. Thalmann. Autonomous virtual agents learning a cognitive model and evolving. In *Lecture Notes in Computer Science*, volume 3631, pages 88–98. Springer-Verlag, Berlin, 2005.

[30] T. Conde and D. Thalmann. Learnable behavioral model for autonomous virtual agents: Low-level learning. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06)*, pages 89–96, Hakodate, Hokkaido, Japan, May 8-12 2006.

[31] M. Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Journal of Nonverbal Behavior*, 26(2):117–139, 2004.

[32] M. de Berg, M. V. Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications (2nd ed.)*. Springer-Verlag, 2000.

[33] P. S. Dodds and D. J. Watts. Universal behavior in a generalized model of contagion. *Physical Review Letters*, 92(21):218701–1–218701–4, 2004.

[34] P. S. Dodds and D. J. Watts. A generalized model of social and biological contagion. *Journal of Theoretical Biology*, 232(4):587–604, 2005.

[35] J. Dollard, L. D. andNeal Miller, H. Mowrer, and R. Sears. *Frustration and Aggression*. Yale University Press, 1939.

[36] F. Durupinar, J. Allbeck, N. Pelechano, and N. Badler. Creating crowd variation with the ocean personality model. In *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1217–1220, Estoril, Portugal, May 12-16, 2008.

[37] F. Durupinar, N. Pelechano, J. Allbeck, U. Gudukbay, and N. I. Badler. The impact of the ocean personality model on the perception of crowds. *IEEE Computer Graphics and Applications*, 99(PrePrints), 2009.

[38] A. Egges, S. Kshirsagar, and N. Magnenat-Thalmann. Generic personality and emotion simulation for conversational agents. *Computer Animation and Virtual Worlds*, 15(1):1–13, 2003.

[39] A. Egges and N. Magnenat-Thalmann. Emotional communicative body animation for multiple characters. In *Proceedings of VCrowds'05*, 2005.

[40] P. Ekman. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Semiotica*, 1:49–98, 1969.

[41] P. Ekman. *Cognition and Emotion*, chapter An Argument for Basic Emotions., pages 169–200. Lawrence Erlbaum Associated Publishers, 1992.

[42] M. S. El-Nasr, J. Yen, and T. R. Ioerger. Flame - fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, 3(3):219–257, 2000.

[43] N. Farenc, S. R. Musse, and E. Schweiss. A paradigm for controlling virtual humans in urban environment simulations. *Applied Artificial Intelligence*, 14:69–91, 2000.

[44] J. Funge. *Making Them Behave: Cognitive Models for Computer Animation*. PhD thesis, Graduate Department of Computer Science, University of Toronto, 1998.

[45] J. Funge. Cognitive modeling for games and animation. *Communications of the ACM*, 43(7):41–48, 2000.

[46] J. Funge, X. Tu, and D. Terzopoulos. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. *ACM Computer Graphics (Proceedings of SIGGRAPH'99)*, pages 29–38, 1999.

[47] P. Gebhard. ALMA a layered model of affect. In *Proceedings of the Fourth International Joint Conference Autonomous Agents and Multi-Agents Systems*, pages 29–36, 2005.

[48] L. R. Goldberg. An alternative "description of personality": The big-five factor structure. *Journal of Personality and Social Psychology*, 59:1216–1229, 1990.

[49] H. H. González-Baños, D. Hsu, and J. Latombe. Motion planning: recent developments. In *Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications*. CRC Press, 2006.

[50] J. Gratch and S. Marsella. Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. In *Proceedings of the fifth international conference on Autonomous agents*, pages 278–285, 2001.

[51] J. Gratch and S. Marsella. A domain-independent framework for modeling emotion. *Cognitive Systems Research*, 5(4):269–306, 2004.

[52] E. T. Hall. *The Hidden Dimension.* Anchor Books, 1966.

[53] Y. Hijikata, T. Komatsu, N. Saiwaki, and S. Nishida. Automatic generation of moving crowd using chaos and electric charge model. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 342–347, 2002.

[54] R. L. Hughes. The flow of human crowds. *Annual Review of Fluid Mechanics*, 35:169–182, 2003.

[55] C. E. Izard. *Human Emotions.* New York & London: Plenum Press, 1977.

[56] D. Jolliffe and D. P. Farrington. Development and validation of the basic empathy scale. *Journal of Adolescence*, 29(4):589–611, 2006.

[57] A. Kamphuis and M. H. Overmars. Finding paths for coherent groups using clearance. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 19–28, 2004.

[58] A. Kamphuis, M. Rook, and M. H. Overmars. Tactical path finding in urban environments. In *Proceedings of First International Workshop on Crowd Simulation (V-CROWDS'05)*, Lausanne, Switzerland, 2005.

[59] Z. Kasap and N.-M. Thalmann. Intelligent virtual humans with autonomy and personality: State-of-the-art. In *Intelligent Decision Technologies.* IOS Press, 2007.

[60] H. Kessler, A. Festini, H. C. Traue, S. Filipic, M. Weber, and H. Hoffmann. *SIMPLEX – Simulation of Personal Emotion Experience, J. Or (ed.)*, chapter 13, pages 255–270. InTech Education and Publishing, 2008.

[61] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.

[62] J. J. Kuffner, Jr. Goal-directed navigation for animated characters using real-time path planning and control. In *Lecture Notes In Computer Science (Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments - CAPTECH'98)*, volume 1537, pages 171–186, November 1998.

[63] Y.-C. Lai, S. Chenney, and S. Fan. Group motion graphs. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 281–290, 2005.

[64] M. S. A. Latif and S. Widyarto. The crowd simulation for interactive virtual environments. In *ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Appications in Industry (VRCAI'04)*, pages 278–281, Singapore, 2004.

[65] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[66] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[67] G. Le Bon. *The Crowd: A Study of the Popular Mind*. Dover Publications, 2002.

[68] K. M. Lean and M. Pasupathi. Collaborative narration of the past and extroversion. *Journal of Research in Personality*, 40:1219–1231, 2006.

[69] S.-H. Lee, H. Pak, and T.-S. Chon. Dynamics of prey-flock escaping behavior in response to predator's attack. *Journal of Theoretical Biology*, 240(2):250–259, 2006.

[70] A. Lerner, E. Fitusi, Y. Chrysanthou, and D. Cohen-Or. Fitting behaviors to pedestrian simulations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'09)*, pages 199–208, New York, NY, USA, 2009. ACM.

[71] T. Li, Y. Ma, Y. Qiu, and P. Yue. Modelling personality, emotion and mood for a pedagogical agent. In *Proceedings of the 25th IASTED International Multi-Conference: Artificial intelligence and Applications (AIAP'07)*, pages 272–277, Innsbruck, Austria, 2007. ACTA Press.

[72] S. Marsella and J. Gratch. EMA: A computational model of appraisal dynamics. In *European Meeting on Cybernetics and Systems Research*, 2006.

[73] S. Marsella and J. Gratch. EMA: A process model of appraisal dynamics. *Journal of Cognitive Systems Research*, 10(1):70–90, 2009.

[74] M. J. Mataric. Learning to behave socially. In D. Cliff, P. Husbands, J. Meyer, and S. Wilson, editors, *From Animals to Animats: International Conference on Simulation of Adaptive Behavior*, pages 453–462, 1994.

[75] R. McDonnell, M. Larkin, S. Dobbyn, S. Collins, and C. O'Sullivan. Clone attack! perception of crowd variety. *ACM Transactions on Graphics (Proceedings of SIGGRAPH'08)*, 27(3):Article no. 26, 8 pages, 2008.

[76] C. McPhail. *The Myth of the Madding Crowd.* Aldine de Gruyter, 1991.

[77] A. Mehrabian. Framework for a comprehensive description and measurement of emotional states. *Genetic, Social and General Psychology Monographs*, 121:339–361, 1995.

[78] A. Mehrabian. Analysis of the big-five personality factors in terms of the pad temperament model. *Australian Journal of Psychology*, 48:86–92, 1996.

[79] A. Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, 1996.

[80] A. Mehrabian. Comparison of the PAD and PANAS as models for describing emotions and for differentiating anxiety from depression. *Journal of Psychopathology and Behavioral Assessment*, 19(4):331–357, 1997.

[81] A. Mehrabian and J. A. Russell. *An approach to environmental psychology.* Cambridge, MA: MIT Press, 1974.

[82] S. Milgram and H. Toch. *The Handbook of Social Psychology, 2nd ed. (G. Lindzey, E. Aronson, eds)*, chapter Collective Behavior: Crowds and Social Movements. Addison-Wesley, Reading, MA, 1968.

[83] R. Millar, J. Hanna, and S. Kealy. A review of behavioural animation. *Computers and Graphics*, 23(1):127–143, 1999.

[84] J.-S. Monzani, A. Caicedo, and D. Thalmann. Integrating behavioural animation techniques. In A. Chalmers and T. Rhyne, editors, *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2001.

[85] W. Morris. *Mood: The Frame of Mind*. Springer-Verlag, New York, 1989.

[86] S. R. Musse, C. Babski, T. Capin, and D. Thalmann. Crowd modelling in collaborative virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST'98)*, pages 115–124, Taipei, Taiwan, 1998.

[87] S. R. Musse and D. Thalmann. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Proceedings of the Eurographics Workshop of Computer Animation and Simulation*, pages 39–51, Budapest, Hungary, 1997.

[88] S. R. Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):152–164, 2001.

[89] A. Ortony, G. Clore, and A. Collins. *The Cognitive Structure of Emotions.* Cambridge University Press, Cambridge, 1988.

[90] M. H. Overmars. Recent developments in motion planning. In *International Conference on Computational Science*, volume 3, pages 3–13, 2002.

[91] R. Parent. *Computer Animation: Algorithms and Techniques.* Morgan Kaufmann, 2001.

[92] S. Paris, S. Donikian, and N. Bonvalet. Environmental abstraction and path planning techniques for realistic crowd simulation. *Computer Animation and Virtual Worlds*, 17:325–335, 2006.

[93] N. Pelechano, J. Allbeck, and N. Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 99–108, 2007.

[94] N. Pelechano, J. Allbeck, and N. Badler. *Virtual Crowds: Methods, Simulation, and Control.* Morgan & Claypool, 2008.

[95] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler. Crowd simulation incorporating agent psychological models, roles and communication. In *Proceedings of First International Workshop on Crowd Simulation (V-CROWDS'05)*, Lausanne, Switzerland, 2005.

[96] N. Pelechano, C. Stocker, J. Allbeck, and N. Badler. Being a part of the crowd: Toward validating vr crowds using presence. In *Proceedings of Autonomous Agents and Multi-Agent Systems*, pages 136–142, 2008.

[97] K. Perlin and A. Goldberg. Improv: A system for scripting interactive actors in virtual worlds. *Computer Graphics*, 29(3), 1996.

[98] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, 21(4), 1987.

[99] C. W. Reynolds. Not bumping into things. *ACM SIGGRAPH'88 Course Notes, #27, Developments in Physically-based Modeling*, pages G1–G13, 1988.

[100] C. W. Reynolds. Steering behaviors for autonomos characters. In *Proceedings of Game Developers Conference*, pages 763–782, San Jose, California, 1999.

[101] N. Saiwaki, T. Komatsu, T. Yoshida, S., and Nishida. Automatic generation of moving crowd using chaos model. In *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, volume 4, pages 3715–3721, 1997.

[102] M. Schreckenber. *Pedestrian and Evacuation Dynamics*. Springer-Verlag, 2001.

[103] S. Shanbhag. Behavioral animation: A report. In *Proceedings of the Inter Research Institute Student Seminar in Computer Science (IRISS'02)*, 2002.

[104] W. Shao and D. Terzopoulos. Autonomous pedestrians. *Graphical Models*, 69(5-6):246–274, 2007.

[105] B. G. Silverman. More realistic human behavior models for agents in virtual worlds: Emotion, stress and value ontologies. Technical report, Systems Engineering Department, University of Pennsylvania, 2001.

[106] B. G. Silverman, G. Bharathy, K. O'Brien, and J. Cornwell. Human behavior models for agents in simulators and games: Part I-enabling science with PMFserv. *Presence: Teleoperators. and Virtual Environments*, 15(2), 2006.

[107] C. Soh, P. Raveendran, and Z. Taha. Automatic generation of self-organized virtual crowd using chaotic perturbation. In *Proceedings of IEEE Region 10 Conference (TENCON'04)*, volume 2, pages 124–127, 2004.

[108] M. Sung, M. Gleicher, and S. Chenney. Scalable behaviors for crowd simulation. *Computer Graphics Forum*, 23(3):519–528, 2004.

[109] M. Sung, L. Kovar, and M. Gleicher. Fast and accurate goal-directed motion synthesis for crowds. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 291–300, 2005.

[110] D. Terzopoulos, X. Tu, and R. Grzeszczuk. Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351, 1994.

[111] D. Thalmann, S. R. Musse, and M. Kallmann. Virtual humans' behaviour: Individuals, groups, and crowds. In *Proceedings of the International Conference Digital Media Futures*, Bradford, UK, April 1999.

[112] B. Tomlinson and B. Blumberg. Alphawolf: Social learning, emotion and development in autonomous virtual agents. In *Proceedings of First GSFC/JPL Workshop on Radical Agent Concepts*, pages 35–45, 2002.

[113] A. Treuille, S. Cooper, and Z. Popovic. Continuum crowds. *ACM Transactions on Graphics (Proceedings of SIGGRAPH'06)*, 25(3):1160–1168, 2006.

[114] R. Turner and L. M. Killian. *Collective Behavior.* Prentice Hall, 1993.

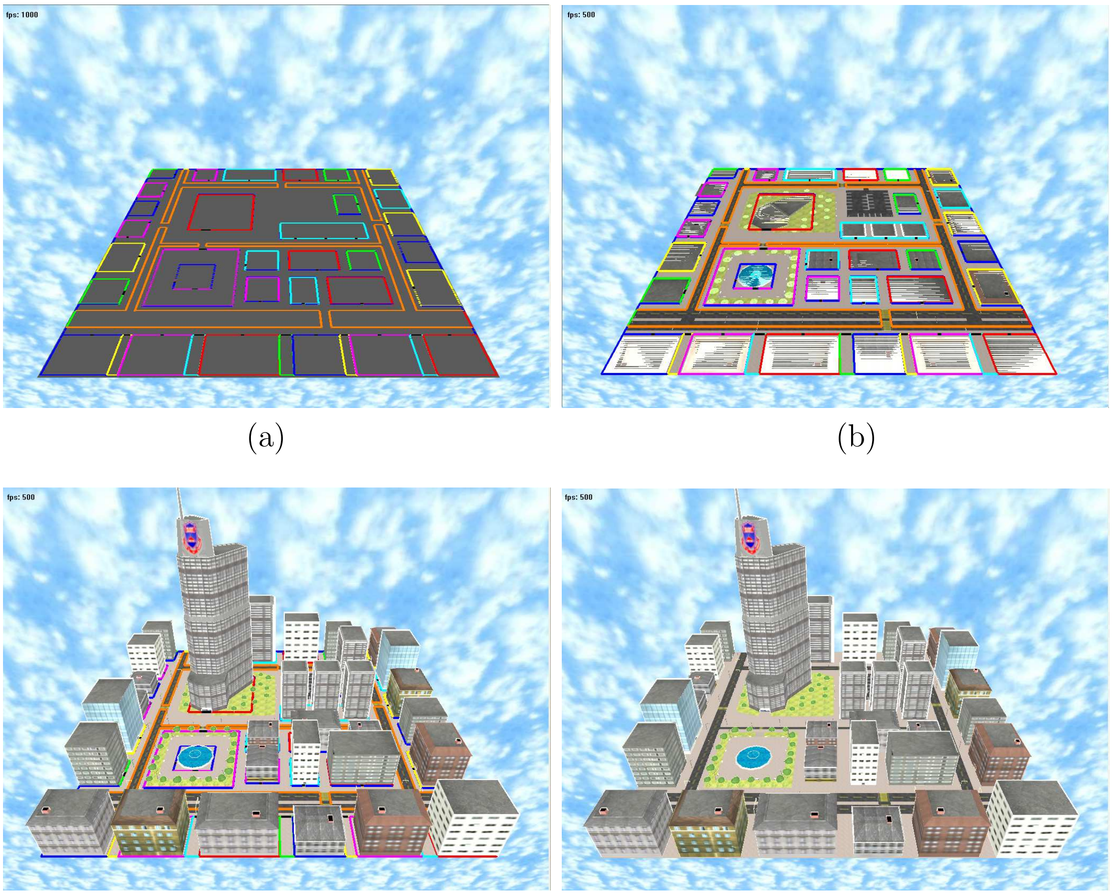
[115] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[116] J. G. Wiggins. *The Five-Factor Model of Personality: Theoretical Perspectives.* The Guilford Press, New York, 1996.

Appendix A

Navigation

Navigation of virtual humans within an environment requires an abstract representation of the navigational space. Computing local motion is not sufficient since agents can get stuck in local minima. Therefore, a more complex path planning methodology is required. HiDAC performs this by creating cell portal graphs (CPG) of the navigation space [94]. HiDAC uses CPGs in indoor environments by extracting a cell portal graph from a special building file. In HiDAC, cells are the rooms and portals are the doors. On the other hand, CPGs can also be used for outdoor environments [4], where cells and portals need to be abstract definitions. We follow the same methodology in our system. Since our scenarios take place outdoors, we create the graphs from the environment model itself. The environment is an .obj file and it just represents the geometry. It does not include any special tagging. Therefore, we need to create the CPG from the model itself. Since HiDAC uses a special purpose building file, instead of creating the CPG from scratch, we first convert our model to the HiDAC building file and then create the CPG using HiDAC's techniques. The floor plan in HiDAC includes horizontal and vertical walls, doors, stairs and obstacles. We also include weak walls. Normally, these are for people falling down and becoming obstacles. However, in our case, we use weak walls to define boundaries of roads. In general, pedestrians only cross the streets through crosswalks. Yet, in case of emergencies, they can cross the streets across the road. Collision rules for weak walls are not as strict as regular walls; agents can just walk through them. Figure A shows the creation of a navigation graph from an environment model of .obj type.



(a) (b)

(c) (d)

Figure A.1: Creating a navigation graph from an environment model, (a) 2D navigation map, (b) 2D navigation map on the projected environment model, (c) 2D navigation map on the environment model, (d) 3D environment model

A building file represents the environment as a grid, showing the discretized locations of walls and portals. The building file is created semi-automatically. It cannot be fully automatic since the model we use for the environment is not tagged and it is not special in any way. Any model file of type .obj can be loaded into the system. Therefore, the program cannot discriminate roads, buildings and entrances of buildings. The program first takes a projection of the environment onto the xz plane. Then, it saves the projected environment to an image file. Next, we run a script that automatically detects the horizontal and vertical lines in the image by edge detection algorithms. These constitute the walls of the building file. The building file is loaded into the system and CPG is automatically generated. Then the user can interact with the program to make certain changes such as adding weak walls, portals or removing unnecessary walls.

In HiDAC, portals are fixed size. We modified the structure to include portals of variable sizes. Normally, the center of a portal is computed as the attractor location when agents need to move from one cell to another. However, we have changed attractor geometry from a point to a line segment. In this case, each agent is attracted to the closest point on the portal. This is performed by taking the agent's projection onto the line segment, which represents the portal (Figure A.2).

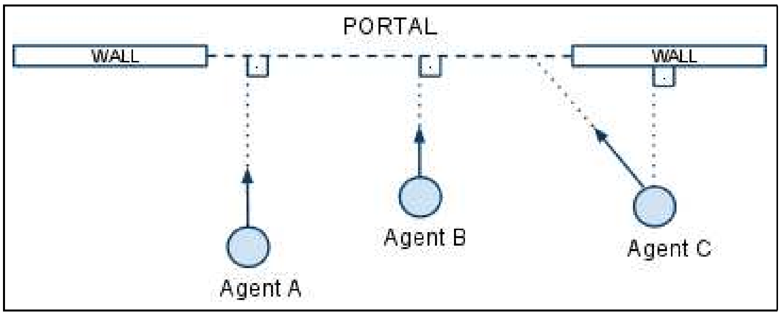


Figure A.2: Agents moving through a linear portal

Appendix B

The System At Work

Our system is is a Single Document Interface (SDI) application implemented using Microsoft Visual C++ 2005 and Microsoft Foundation Classes (MFC). The graphics display API OpenGL is used. The top level user interface of the system is seen in Figure B.1. The elements on the interface can be mainly divided into three parts:

1. *Main Menu*: This consists of menu bar and toolbar. It basically allows the user to control the application.
2. *Control Toolbox*: This toolbox allows the user to create crowds in various scenarios, change the underlying psychological parameters of crowds, modify drawing settings and create and modify the navigation map of the environment. It consists of four panels: *Crowd, Psych, Control* and *Environment*.
3. *Viewing Area*: The viewing area shows the perspective or orthogonal view of the 3D environment.

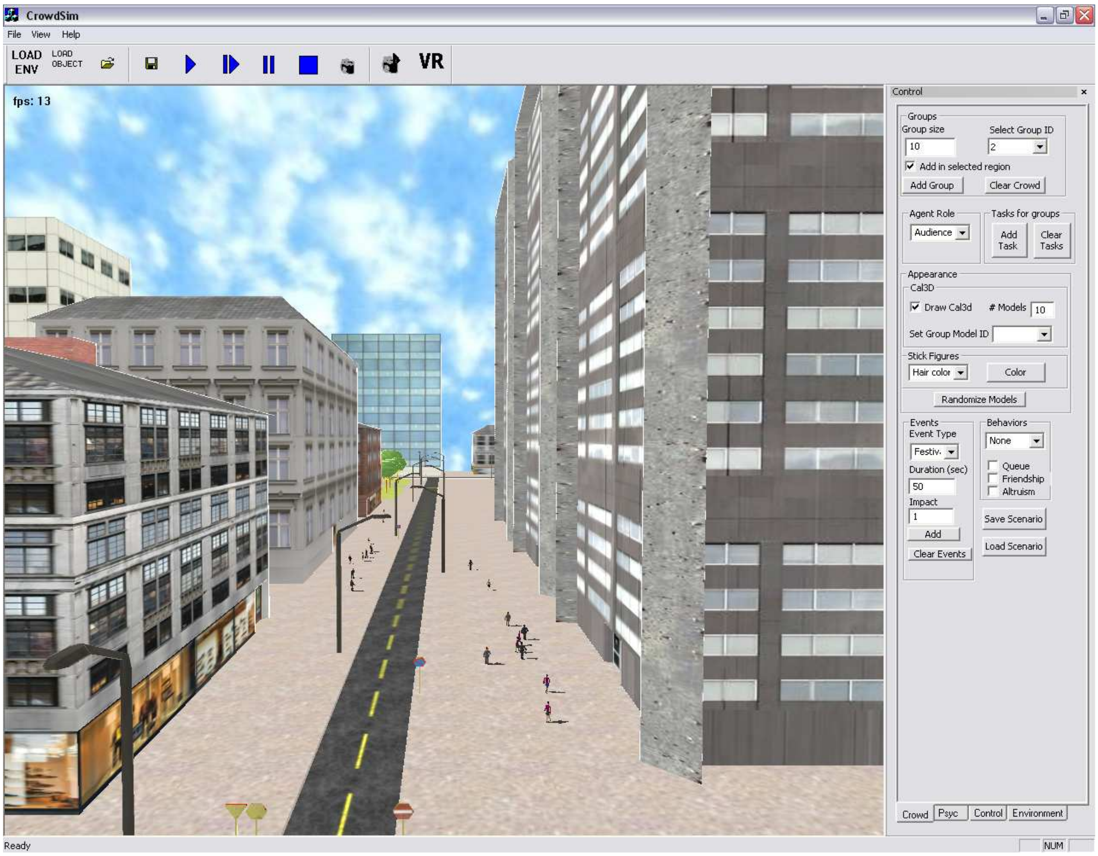


Figure B.1: Top level user interface of the system

The main menu part of the program consists of the menu bar and the other toolbars. The menu bar includes “File”, “View” and “Help” subitems and provides the general functionalities like loading an environment model or an object model, changing the user interface options, and giving information about the program. The user also can start, stop, pause and step by step run the animation by using the toolbar. The toolbar gives user the opportunity to record the animation or take a snapshot of it. The VR mode allows the user to see the environment through the eyes of an agent in the simulation.

Control toolbox includes four panels. The main control of the simulation is handled through the crowd panel. The user can create groups of people with different characteristics and purposes, load 3D models for the virtual humans’ rendering and animation. Group size is also determined by the user. As well as the characteristics of the individuals in the crowd, the user can select from various scenarios such as festival or explosion. The system also enables the user to save the current scenario or load an existing one. Psych panel, as the name suggests, enables the control of the psychological traits of the selected groups. The user can set the means and standard deviations of any of the personality, mood or emotion parameters. Control panel lets the user enable or disable some underlying simulation variables such as the 2D view of the environment, cell portal graphs, shadows, or task locations. Finally, environment panel facilitates the user to create the navigation graph for the existing environment file. In addition, the user can add several objects to the scene through this panel. Figure B.2 shows each of these panels. The keyboard and mouse controls are presented in Table B.

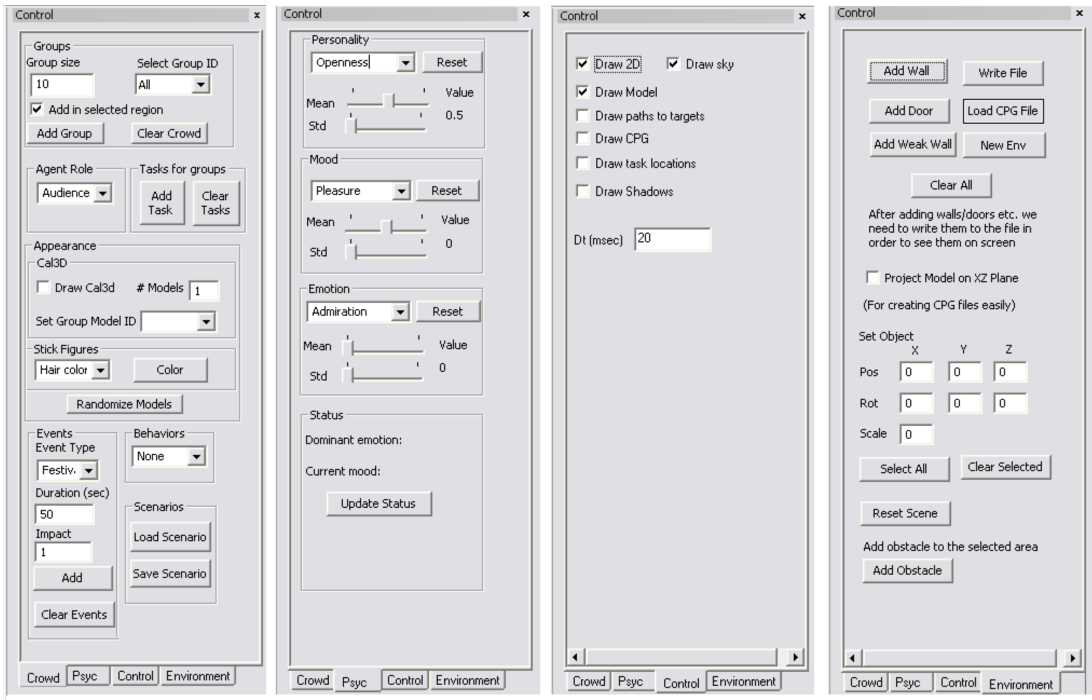


Figure B.2: The control toolbox

Buttons	Controls
Up	Moves forward in VR mode
Up	Translates the selected object in +y direction in 3rd person mode
Down	Moves backward in VR mode
Down	Translates the selected object in -y direction in 3rd person mode
Left	Moves right in VR mode
Left	Translates the selected object in -x direction in 3rd person mode
Right	Moves left in VR mode
Right	Translates the selected object in +x direction in 3rd person mode
Home	Translates the selected object in +z direction in 3rd person mode
End	Translates the selected object in -z direction in 3rd person mode
Page Up	Rotates the head up in VR mode
Page Down	Rotates the head down in VR mode
+	Increases speed in VR mode
-	Decreases speed in VR mode
1	Scales down the selected object
2	Scales up the selected object
W	Rotates the selected object clockwise around the x axis.
S	Rotates the selected object counterclockwise around the x axis
A	Rotates the selected object clockwise around the y axis
D	Rotates the selected object counterclockwise around the y axis
Z	Rotates the selected object counterclockwise around the z axis
X	Rotates the selected object counterclockwise around the z axis
R	Resets the viewpoint
Left Mouse Click	Selects a point on the ground or selects an obstacle
Left Mouse Drag	Selects a region on the ground
Left Mouse	Applies user force
Right Mouse Click	Deselects the point or region
Right Mouse Drag	Zooms the camera in/out
CTRL + Left Mouse Drag	Rotates the camera
CTRL + Right Mouse Drag	Translates the camera
Shift + Left Mouse Drag	Translates selected object
SPACE	Toggles between perspective and orthogonal top views

Table B.1: Keyboard and mouse controls in the system