COMMUNICATION MODELS FOR CROWD SIMULATION

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

By
Kurtuluş Küllü
July 2017

Communication Models for
Crowd Simulation
By Kurtuluş Küllü
July 2017

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Uğur Güdükbay(Advisor)

Varol Akman

Selim Balcısoy

Özgür Ulusoy

Faruk Polat

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

COMMUNICATION MODELS FOR CROWD SIMULATION

Kurtuluş Küllü
Ph.D. in Computer Engineering
Advisor: Uğur Güdükbay
July 2017

Modeling and animation of behaviorally plausible virtual crowds are important problems of crowd simulation research. We propose a communication model in order to equip virtual agents with the ability to autonomously communicate with each other. We investigate whether such a communication model would improve the plausibility of the simulated crowds. Initially, our efforts were towards a model that is as human-like as possible and towards combining this model with an agent architecture that contains psychological attributes. Early experimental results showed that when we look at a crowd, the influences such as different agent personalities causing different communicative behavior are hardly visible. Besides, achieving these effects introduces complexity. Thus, a generic and easy-to-use communication model instead of a human-like one became the target and psychological agent attributes were dropped.

The proposed communication model and its application in several scenarios are presented in this dissertation. As a second contribution, one of the application scenarios led us to develop a planning algorithm for an agent in an unknown environment. Simulation results are analyzed both visually and by using various measurements and metrics. Our conclusion is that in addition to improving observed behavioral variety, the effects of employing the communication model are clear in the quantitative results and these effects are in line with our expectations in each scenario.

Keywords: Crowd simulation, communication model, agent communication, Foundation for Intelligent Physical Agents (FIPA), Agent Communication Language (ACL).

ÖZET

KALABALIK SİMÜLASYONLARI İÇİN İLETİŞİM MODELLERİ

Kurtuluş Küllü
Bilgisayar Mühendisliği, Doktora
Tez Danışmanı: Uğur Güdükbay
Temmuz 2017

Davranışsal olarak inandırıcı sanal kalabalıkların modellenmesi ve canlandırılması, kalabalık benzetimi araştırmalarının önemli problemleridir. Tez çalışmamızda, sanal bireyleri, otonom olarak birbirleriyle iletişim kurma yeteneği ile donatmak için bir iletişim modeli önerilmiştir. Böyle bir iletişim modelinin, benzetimi yapılan kalabalıkların inandırıcılığını artırıp artırmayacağı araştırılmıştır. Başlangıçta çabalarımız, mümkün olduğunca insana benzer bir modele ve bu modeli psikolojik nitelikler de içeren bir sanal birey mimarisiyle birleştirmeye yönelikti. Erken deneysel sonuçlar, kalabalığa baktığımızda, farklı sanal birey kişiliklerinin farklı iletişimsel davranışlara neden olması gibi etkilerin fazla görünür olmadığını göstermiştir. Ayrıca, bu etkilerin başarılması karmaşıklığı artırmaktadır. Bu yüzden, insan benzeri bir iletişim modeli yerine genel ve kullanımı kolay bir iletişim modeli hedeflenmiştir ve psikolojik sanal birey niteliklerinin modellenmesinden vazgeçilmiştir.

Önerilen iletişim modeli ve çeşitli senaryolarda uygulanmasının sonuçları bu tezde sunulmaktadır. İkinci bir katkı olarak, uygulama senaryolarının birinde ihtiyaç duyulması sonucunda, bir sanal bireyin bilmediği bir ortamda planlama yapmasını sağlayan bir algoritma geliştirilmiştir. Benzetim sonuçları hem görsel olarak hem de çeşitli ölçümler ve metrikler kullanılarak analiz edilmiştir. Vardığımız sonuç, gözlenen davranışsal çeşitliliğin iyileştirilmesine ek olarak, iletişim modeli kullanıldığında etkilerin sayısal sonuçlarda da gözlemlendiği ve bu etkilerin gerçekleştirilen senaryolarda beklentilerimizle uyumlu olduğu yönündedir.

Anahtar sözcükler: Kalabalık simülasyonu, iletişim modeli, aracı iletişimi, Akıllı Fiziksel Aracılar Kuruluşu (İng. FIPA), Aracı İletişim Dili (İng. ACL).

Acknowledgement

To begin with, I would like to express my sincere gratitude to my advisor Prof. Uğur Güdükbay for his guidance, patience, motivation, and support throughout my Ph.D. studies. I would also like to thank Prof. Dinesh Manocha for enabling my visit to the University of North Carolina at Chapel Hill and for all his guidance and invaluable contributions.

I would like to thank the members of the monitoring committee and defense jury Prof. Varol Akman, Assoc. Prof. Selim Balcısoy, Prof. Özgür Ulusoy, and Prof. Faruk Polat.

I am grateful to my dear wife Pınar and our precious daughter Arya, whose existence itself is the greatest inspiration and support. I also cannot thank the rest of my family and friends enough for the support they always provide.

I would like to express my appreciation to all my mentors and colleagues at Bilkent, Ankara University, the University of North Carolina at Chapel Hill and other institutions. Special thanks go to İ. Sengör Altıngövde, Aytek Aman, Ateş Akaydın, K. Barış Atıcı, Ö. İlker Küçüktepe, Yılmaz Ar, and G. Erkan Bostancı. I also wish to acknowledge Architect Mete Sezer, who willingly shared the graphical computer model of his building design.

Last but not least, I would like to take the opportunity to thank Nobel Laureate Prof. Aziz Sancar and Prof. Gwen Sancar not only for hosting me and my family at Chapel Hill but also for setting an inspirational and motivational example for us.

This research was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant No. 112E110. Additionally, I was supported by a scholarship (support type 2214-A) by TÜBİTAK to visit the University of North Carolina at Chapel Hill during my Ph.D. studies.

Contents

List of Figures

List of Tables

List of Publications

This dissertation is based on the following publications. The rights to use the whole content of these publications in this thesis are obtained from the publishers.

1. K. Kullu and U. Güdükbay. A Layered Communication Model for Agents in Virtual Crowds. In Proceedings of 27th International Conference on Computer Animation and Social Agents (CASA'14), Short Papers, Houston, USA, May 2014.

2. K. Kullu, U. Güdükbay, and D. Manocha. ACMICS: an agent communication model for interacting crowd simulation. Journal of Autonomous Agents Multi-Agent Systems, doi:10.1007/s10458-017-9366-8, Springer, 2017.

Chapter 1

Introduction

Virtual crowd simulation is a research topic related to the fields of computer graphics and artificial intelligence (AI). The simulated crowd consists of individuals that are often called agents, which can stand for people or other entities. Crowd simulation algorithms and the resulting simulations are commonly used in virtual environments. They are also widely used to generate plausible effects in computer animation and games. Other areas of application are the prediction of pedestrian movement in evaluating structural and urban designs [1] and navigation of robots among people [2].

A simulation's similarity to the corresponding real-world event is called *fidelity*. A high-fidelity simulation reproduces a real-life scenario better than a low-fidelity one. This does not mean that high-fidelity simulations are always more useful [3]. Especially for simulations with entertainment and educational purposes, low-fidelity simulations can produce better focus and results. However, for some simulations, such as those targeting safety evaluation, high-fidelity is a desired property. As a result, automatically simulating behaviorally plausible crowds is one of the primary aims of the research in this field. The plausibility is often achieved by improving heterogeneity of the crowd and agents displaying emergent behaviors [4, 5]. There are numerous works [4, 6, 7, 8, 9] studying these points and suggesting cognitive, behavioral, and psychological models for improvement.

1.1 Motivation

Increased fidelity through behavioral realism is particularly important for safety engineering applications. Human population grows every day and crowds are a part of daily life, especially in large cities. Despite regulations about buildings, emergency situations, and event organizations, recent history is full of tragic events involving crowds. The Hillsborough Stadium crush in England in 1989 (96 casualties), the 9/11 attacks on U.S. in 2001 (near 3000 casualties), the Station nightclub fire in Rhode Island in 2003 (100 casualties), the Love Parade disaster in Duisburg in 2010 (21 casualties), the Sewol Ferry Disaster in South Korea in 2014 (approximately 300 casualties), the Mina stampede in Mecca during the Hajj in 2015 (over 2000 casualties), and the recent Grenfell Tower fire in London (over 80 casualties) are only some well-known examples.

A common characteristic of all these incidents is that loss of lives did not happen at an instant but over a time interval. As a result, whether lives could have been saved with better crisis management is a question that is always asked. For example, the Grenfell Tower residents remained in their homes because of a 'stay put policy', which is now being questioned. Behaviorally realistic crowd simulations can help in foreseeing such problems so that precautions can be taken before lives are actually lost.

This dissertation is motivated by a simple observation: Individuals in real-world crowds communicate. Consider concert or sports event spectators or people evacuating a building. The information shared between the individuals can naturally influence the behavior and movement of the crowd. For example, an evacuating person can talk with others nearby, or read signs; a herd member can warn the others about a predator; a driver should use signals to inform others when turning or changing lanes. In some cases, a leader in a crowd, such as a police chief or a fire marshal, can inform the people to manage an incident, such as guiding the evacuation from a dangerous location. These examples show that communication can be important for an individual in a crowd in order to make decisions. Despite this fact, information exchange between the agents has not received much attention in the crowd simulation field. Our work mainly deals with modeling deliberate inter-agent communication in virtual crowd simulations and analyzing the effects of such communication to the overall crowd behavior.

Focusing only on deliberate communication is necessary because taking communication as a broader concept makes our task very difficult, if not impossible. If a broader definition of communication, such as "transfer of information", is used, then almost everything can be thought as "communication". For instance, perception can be considered as information being transferred from the environment to the agent [10], and therefore, perception can be regarded as one form of communication. To prevent such complexities, we focus on deliberate inter-agent communication. Overall, we do not take into account phenomena that can be considered as unintentional communication. Yet, an agent following other agents, which, in an informal context, commonly regarded as unintentional/indirect communication, is taken into account as part of the agent navigation.

1.2 Contributions of the Thesis

A novel way to simulate inter-agent communication in the context of virtual crowd simulation is proposed. The impact of communication on the behavior of the simulated crowd is evaluated. A simplified adaptation of a message structure specification from the multi-agent systems (MAS) community, known as Foundation for Intelligent Physical Agents (FIPA) Agent Communication Language (ACL) Message Structure Specification [11], is used. FIPA is a standards organization operating under IEEE, which aims to produce software standards specifications for agent-based systems and MAS. The proposed communication approach can handle human-like, inter-agent message exchange in a virtual crowd. We make no assumptions about agents' (local or global) navigation capabilities; our approach can be combined with any preferred navigation implementation. The novel contributions of our work are:

1. A model to facilitate inter-agent communication in a crowd simulation system that
   (a) is designed as a separate module in the agent architecture,
   (b) requires some form of perception capability,
   (c) separates low- and high-level tasks in a modular manner, and
   (d) can be easily extended and used in arbitrary scenarios and/or can support different forms of communication.
2. A high-level planning algorithm to simulate the evacuation behavior in new or unknown environments where the agents do not have a priori knowledge about their environment. We demonstrate that the agents autonomously communicate to navigate more effectively in such scenarios based on our communication model.

Applications of the communication model in various scenarios to facilitate deliberate inter-agent communication are provided in this dissertation. First, it is used in facilitating hollow communications, i.e., communications with no important information transfer. We measured the pedestrian flow both with and without communication and compared the results with those from other simulators. Then, we combined the communication approach with the high-level evacuation planning algorithm so that its application in enabling meaningful agent interactions is highlighted. We analyzed the effects on pedestrian evacuation times and trajectories. Finally, a comparison using the vfractal metric [12] is provided between trajectories extracted from a real crowd video and simulated agent trajectories with/without communication. The proposed communication model can be combined with any crowd simulation method and does not significantly increase the complexity. Crowds consisting of tens or hundreds of agents can practically be simulated at interactive rates on current desktop systems.

1.3 Organization of the Thesis

The organization of the rest of the dissertation is as follows. In the next chapter, a comprehensive summary of related literature on crowd simulation, communication models in general, communication of virtual agents, and the vfractal metric is provided. The agent architecture is described next in Chapter 3. Then, the communication model is discussed and explained in Chapter 4. In Chapter 5, various preliminary experiments that led to a change of direction in our work are discussed. We used the scenarios described in Chapter 6 in evaluating our final model and highlighting the performance. Lastly, we conclude and discuss the limitations of our work and possible future extensions in Chapter 7.

Chapter 2

Related Work

As described in Introduction, our primary aim is to incorporate a communication model for autonomous agents in a virtual crowd. There are several efforts for which the main aim is to give a computer controlled autonomous agent the ability to communicate. We will discuss some of these works in detail but before doing so, we would first like to provide a summary of crowd simulation and communication literatures.

2.1 Crowd Simulation

We found two comprehensive overviews of crowd simulation algorithms, one by Ali et al. [13] and the other by Thalmann and Musse [14]. Pelechano et al. [5] provide another survey of common crowd simulation methods and existing crowd simulation algorithms and systems. Additionally, the authors' three part simulation system (CAROSA + HiDAC + MACES) is explained to a certain detail.

In general, virtual crowd simulation approaches are commonly grouped into two classes: *macroscopic* and *microscopic*. The focus of macroscopic approaches is not the individuals in a crowd but the crowd as a whole. In an opposing sense, the behaviors and decisions of individuals, as well as their interaction with each other, are considered more important in microscopic approaches. Other classifications with names such as fluid-dynamic or gas-kinetic models [15], social force models [16], cellular automata models [17], velocity-based methods [18], and biomechanic models [19] are also used.

There has been a need to populate virtual environments with crowds for movies and animations. The "boids" (bird-oid objects) by Reynolds [20] are one of the earliest and most popular solutions to this need. For movies and animations, crowd simulation can be accomplished by offline techniques. More recently, a similar need has emerged for games and virtual or augmented reality environments but this time there is a real-time constraint [21]. Additionally, there is a desire for agents to react to events in real-time which calls for agent autonomy.

Some crowd simulation efforts [22, 23] concentrate on producing visually plausible crowds whereas others [24, 4, 25, 26] concentrate on producing behaviorally plausible crowds [27]. There are many cases in which a concept from psychology or cognitive literature is applied to simulating crowds. For instance, three elements, namely, *personality*, *emotion*, and *mood*, are incorporated into an agent model in [6]. Turkay et al. [28] use information theoretical concepts in building analytical maps. Then, they use these maps in adaptively controlling agent behavior. All such efforts aim to enhance the plausibility of the crowd with adding as little complexity as possible. Kim et al. [9] claim that a common property of methods using such behavioral or psychological models is that dynamic behavior changes are not directly possible. Based on this observation, they suggest a method to achieve such dynamic behavior changes. Their algorithm is based on General Adaptation Syndrome and models dynamic behaviors in a virtual crowd. The algorithm involves simulation events called *stressors*. Agents accumulate stress from these stressors and their decision-making processes are affected by the level of stress.

In [29], Silverman and colleagues make use of performance moderator functions (PMFs) from behavioral literature to improve the realism of socially intelligent agents. Their overall system, called *PMFserv*, is claimed to integrate PMFs on physiology and stress; personality, cultural and emotive processes; perception;

social processes, and cognition. The second part of their work described in [30] consists of integrating the PMFserv framework into a commercial game engine to test whether it can improve the realism of autonomous characters.

Jaros et al. [27] analyze the behavior of real pedestrians in a train station and use their observations to create a three-level behavioral model for virtual pedestrians in a similar environment. The aim of their simulations is to provide a testing environment for building designers to evaluate space utilization. Narain et al. [31] propose a hybrid method to solve the collision avoidance problem for dense crowds in a scalable fashion. Their continuum-based method makes it possible to simulate a hundred thousand agents at near-interactive rates. Qiu and Hu [32] present a framework to model group behaviors in a simulated crowd. Their model enables easy modeling of different types of group structures and their experiments show that different group sizes, intra-group structures, and inter-group relationships significantly influence crowd behaviors.

The Agent Development and Prototyping Testbed (ADAPT) [33] and Menge (German for "crowd") [34] are two recent extensible modular frameworks aimed at simulating virtual agents. ADAPT framework includes facilities for character animation, navigation, and behavior. Its primary focus is on animation, in particular, on the seamless integration of multiple character animation controllers. It couples a system for blending arbitrary animations with static (path finding) and dynamic (steering) navigation capabilities for human characters. It also includes an authoring structure so that new behavior routines can be integrated.

The primary focus of ADAPT is animation. In order to allow multiple animation controllers act on a character simultaneously, they are implemented as modular components, called *choreographers*, each of which works on its own invisible copy of a character skeleton (*shadow*). A coordinator performs a weighted blend of choreographers to produce a final pose.

Navigation is handled with navigation meshes and a predictive goal-directed collision avoidance mechanism. Controlling agent behaviors is achieved via parametrized behavior trees (PBTs) [35]. When they are used in a centralized way, they also allow coordination of multiple agent interactions. PBTs invoke agent's navigation and animation capabilities like `GoTo(), GazeAt()` with three-dimensional (3D) space positions as parameters.

Menge, developed by Curtis et al. [34], is another extensible modular framework but here, the primary focus is on crowd movement. The crowd simulation problem is decomposed into four subproblems, each of which is to be solved for every agent in the crowd: *goal determination*, *planning*, *facilitating reactive behavior*, and *agent motion*. Although these names suggest general abstractions of possibly complex concepts, Menge's concentration on crowd movement reduces them to different stages of motion planning. For example, given a goal for an agent which is to go to a particular position, second subproblem (planning) reduces to finding a path avoiding static obstacles and third subproblem (reactive behavior) reduces to following that path avoiding dynamic obstacles. In addition to the built-in implementations of existing solutions for each subproblem, new solutions can be integrated into the framework via its plug-in architecture.

When using Menge, the details of a scenario can be specified as an Extensible Markup Language (XML) document. Two options are available for visualizing Menge's result. The first one is a simple, interactive 3D visualizer included with the package. The other option is to export agent trajectories and behaviors for use with external visualizers.

Agent behavior is modeled with Behavioral Finite State Machines (BFSMs) that govern goal determination and planning subproblems. States in a BFSM are referred to as FSM-states, which are different from agent states. Menge considers two parts to an agent's state. The position and the velocity of an agent together forms the a-state (agent state) and the collection of all other properties of an agent is called the b-state (behavior state). This is, again, an indication (and a result) of the fact that the main problem being tackled is motion planning. Agents are independent entities in Menge, i.e., centralized agent controls are mostly avoided. The only major centralized part is the handling of spatial queries (such as proximity checks). Simulations are parallelized at the agent level. The efficiency of the parallelization method is evaluated in [36].

Both frameworks are modular and extensible but there are some important differences. ADAPT focuses on skeletal animation blending while Menge's focus is on agent trajectories. Also, due to the visualization complexities preferred and ADAPT's focus on articulated character animation, the number of agents it can simulate at interactive rates is much less than Menge's. ADAPT is reported to achieve interactive frame rates approximately up to 150 agents. Most of the computational cost result from the animation system (choreographers' complexity in particular). The Menge technical report lists some example simulations in one of which (battle simulation) 32000 agents are simulated.

2.2 Communication

Communication is an extremely broad area that is often studied from the perspectives of other disciplines. It is claimed in [37] that although there is a large body of literature and investigation about communication, an identifiable field of communication theory does not exist. Yet, we are provided with various communication models.

Shannon and Weaver developed one of the earliest communication models [38]. Their aim was to mirror the workings of radio and telephone technologies. There were four main components to their initial model: *sender*, *channel*, *receiver*, and *noise*. Many researchers, such as Berlo [39], later extended this initial model. These are generally called *transmission models* of communication. Eight components are commonly considered in transmission models:

- Source;
- Message;
- Transmitter;
- Signal;
- Channel (Carrier);

- Noise;
- Receiver; and
- Destination.

The advantages of transmission models are that they are simple, general, and quantifiable. They have been the base for and heavily used in telecommunications. However, when the aim is to model real-world human communications they are considered inadequate. Some of their inadequacies are listed by Chandler in [40]. For example, one inadequacy is that they do not allow for differing interpretations.

A different view of communication was provided by Wilbur Schramm [41], whose works are mainly related to mass communication and its effects. This view particularly indicates that desired or undesired impact of messages on the target should be examined. In some sense, Schramm's model tries to incorporate human behavior into the communication process.

In Schramm's model, communication is composed of at least three components: *source* (individual, publishing house, and so on), *message* (in the form of ink on paper, sound waves, and so on), and *destination* (such as an individual listening or reading, or lecture audience). In order to communicate, the source encodes her/his message, i.e., puts the information (or feeling) into a form that can be transmitted. A key observation at this point is that the message is independent of the source once it is encoded and sent. The message must be decoded for the communication to be complete.

A communicator can be both an encoder and a decoder, i.e., can both transmit and receive. A communicator receives a signal in the form of a sign. If that sign was learned, then certain responses was also learned with it. Schramm calls these responses mediatory responses and takes them as the meaning of the sign for the individual. Mediatory responses are learned but they are also affected by the current state of the individual. These responses, in turn, trigger learned reactions again subject to the current state. Meaning extracted from decoding induces a new encoding process. What is encoded depends on available response

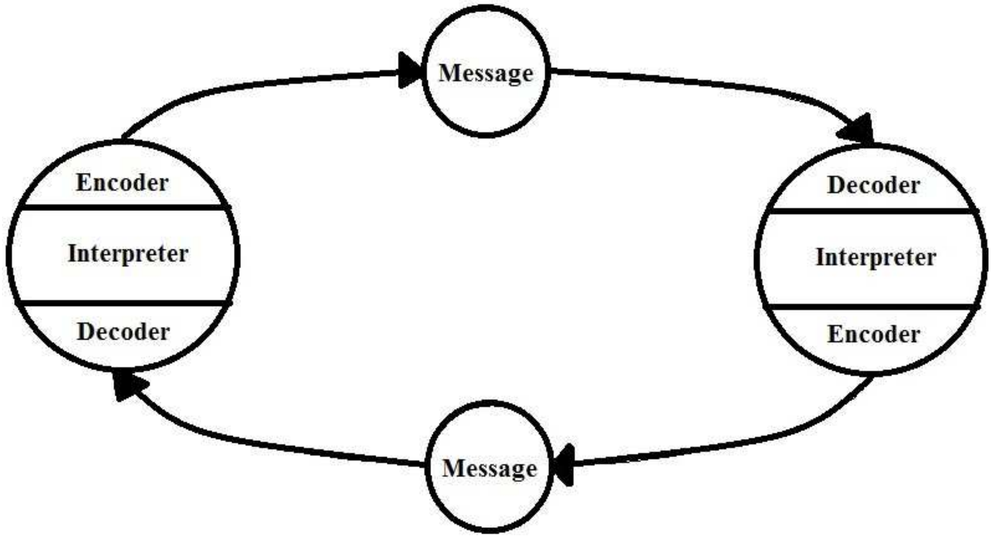


Figure 2.1: The reciprocal model of communication, redrawn from original in [41].

choices suitable for the situation that the individual is in. This encoding in return can result in a new communication or an action. This circular process that the individuals are constantly engaged in is shown in Figure 2.1.

The description up to this stage is in line with the model of communication by Shannon and Weaver but, Schramm's model introduces two major novel concepts: *feedback* and *field of experience (FoE)*. The return process of the circular view of communication described in the previous paragraph is called feedback. This tells an individual how her/his message is interpreted. A communicator often modifies her/his messages according to what is decoded from the feedback. The FoE is a representation of a communicator's beliefs, values, and experiences as well as learned meanings both as an individual or part of a group.

A strength of the model by Schramm is the concept of FoE. The justification for this concept is the intuitive fact that a receiver and a sender must be in tune for communication to be successful. The concept is not very crucial for telecommunication since receivers and senders are often devices that can be tuned (e.g., a radio receiver can be set to the same frequency with the transmitter). On the other hand, it is complicated when we are considering human communication.

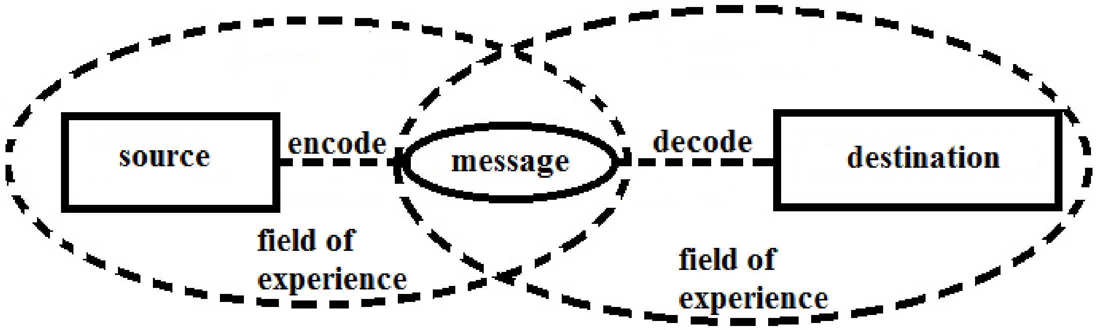


Figure 2.2: Fields of Experience in communication, redrawn from original in [41].

Figure 2.2 is a redrawing from Schramm's original article. The ellipses drawn around represent the accumulated experience of the two individuals trying to communicate. An individual can encode or decode only in terms of her/his own accumulated experience. A classic example would be the primitive tribesman thinking that the camera device captures and imprisons the soul. If a person has never seen, heard, or read about a camera or something similar, and in addition (s)he sees one functioning, (s)he can initially only decode with respect to her/his experiences in life. If the ellipses do not meet, then communication will not be possible. The larger the intersection area is, the easier communication becomes. Schramm's model is much more human communication oriented compared to Shannon and Weaver's model. It considers even more complex issues than what is briefly summarized here such as feedback from one's own messages and multi-channel message sending.

A more recent model developed by Barnlund [42] emphasizes parallelism in the communication process. In this model, communication is modeled as simultaneous sending and receiving of messages (and/or feedback). Models summarized above and similar ones are often challenged particularly from social sciences perspective. Some arguments used to argue that communication is much more complex are:

- How one communicates determines the interpretation of the message.
- Information is separate from the communication itself.

- Personal filters of sender and receiver can vary (different cultures, genders, and so on.)

Communication is an extremely broad area that is often studied from perspectives of different disciplines. Especially when we delve into disciplines like psychology or social psychology, the abundance of related theories become overwhelming for someone from another discipline. In his highly cited work [37], Robert Craig claims that communication theory as an identifiable research field has not been established. He bases this claim on the following arguments:

- Researchers operate on separate domains.
- Books and articles on communication theory seldom mention others except within narrow (inter)disciplinary specialties and schools of thought.
- No general theory or no common goals exist to which all refer.

In order to support these arguments, he denotes Anderson's work [43], which analyzed seven communication theory textbooks and found that they contained 249 distinct theories 195 of which appeared in only one book (just 22% appeared in more than one book). Moreover, only 18 of 249 (7%) were included in more than three books. In addition to his claims emphasizing the problem, he suggests how a unified, central theory of communication can be established. He also sketches what he calls seven traditions of communication theory briefly as follows:

- *Rhetorical*: practical art of discourse;
- *Semiotic*: intersubjective mediation by signs;
- *Phenomenological*: experience of otherness, dialogue;
- *Cybernetic*: information processing;
- *Sociopsychological*: expression, interaction, and influence;
- *Sociocultural*: (re)production of social order;
- *Critical*: discursive reflection.

2.3 Communication and Virtual Agents

Embodied Conversational Agents (ECAs) is a field of research that is concerned both with communication and virtual agents [44]. ECA research tries to address all aspects of conversation because the main aim is to develop computer controlled agents that can carry out a conversation directly with a human. There are two popular approaches to developing such agents: *linguistics oriented* and *animation oriented* [45]. Linguistics oriented approach focuses on the content and form of the conversation, while animation oriented one focuses on facial expressions and gestures. In the first group, the primary concern is what the agent is saying, both words used and underlying meaning. Efforts in the second category are also valuable because actions such as head or eye movements are important components of conversations in real life. Animating these kinds of behaviors for virtual agents greatly improves the realism of a simulation. The works on ECAs are not directly related to our work because they focus on agents communicating with real people through specific forms such as speech, whereas our focus is on inter-agent communication not in a specific form but at an abstract level.

A framework to distribute dialogs among virtual crowd agents is presented in [45]. Their virtual crowd simulations involve (unscripted) conversations that are initiated and guided by agents' attributes and the environmental context. A set of scenario dependent conversational archetypes, such as simple asking-answering, friendly chatting, bargaining, and arguing, is used in a three stage dyadic conversation model. In addition, situation types (postural state of agents when starting a conversation), relationship types (family members, friends, strangers, etc.), and a representative set of environment contexts (street, restaurant, library, etc.) are also used. Five classes of agent attributes facilitate agent heterogeneity. These attributes are grouped as static (age, gender, personality, culture), temporal (calendar), relational (friends, family members, coworkers, seller, customer, supervisor, teacher-student), dynamic (emotion and mood), and behavior and constraint (hands are occupied with a cell phone or coffee). Our work differs from this one in that the dialogs here are only for visual improvement and they do not include information sharing that could affect behavior.

The three part system (CAROSA + HiDAC + MACES) by Pelechano et al. [5] includes is a simple communication model between agents. It is applied in an evacuation scenario and the communication capability is as follows. A Cell-Portal Graph (CPG) structure captures the connectivity of the building to be evacuated. Some agents, called trained agents, have complete information about the building (i.e., they have the full CPG), whereas others know only parts they visited (i.e., they have a subgraph of the full CPG). Agents' communication is modeled as a partial sharing of their subgraphs. The sharing is partial in order to make the agents' behavior closer to real humans. The authors preferred to limit information sharing to two levels of adjacency from current cell because people in real life are unable to give detailed information about all of the structure. Also, a panicking agent may get disoriented and lose part or whole of its information about the environment.

Partial information sharing and panicking agents obviously make the communication more realistic, but the amount of information exchanged between agents can still be considered as static with respect to communicating agent's knowledge. Agents can have different amounts of environmental knowledge but they are always capable of communicating a standard part of it. Whether we could make this amount dynamic was a question that led us to this dissertation. A stressed agent in the real world may not be able to communicate efficiently. It may even be the case that an agent gives incorrect information due to high levels of stress. As a result of these considerations, we were curious whether an agent-based simulation model could be established in which agents' attributes influence their communication capability.

Oijen and Dignum designed a system in which believable human-like communication could be established [46]. In this work, the agents exist in a MAS cognitively. A model for effective agent communication at the cognitive level is proposed. While communicating intents, agents also display believable behavior. The communication signal types are not restricted in the model. Some possible types are speech acts, meta-conversational signals (e.g., turn taking), and affective signals like emotional state. It is argued by the authors that if the aim is to simulate human-like agent interactions, then using MAS standards such as FIPA

ACL are not adequate. They introduce an intermediate layer that coordinates the MAS's capabilities and services with a game engine produced virtual environment. Therefore, separating the mind and body of an agent is the main concern of this work. This separation allows agents to express behaviors, interpret intents of others, and monitor and interrupt scheduled communication.

Park et al. [47] consider formation and sustainability of small groups in a larger crowd to enhance realism. The cohesiveness of these small groups is sustained with member agents' communication. Using this common ground theory based simulation and user studies, the authors show that the animation plausibility is improved through agents' communicative and social interactions.

Henein and White [48] fuse simulation of human factors such as information discovery and communication with a cellular automata crowd model. A large-room evacuation scenario that includes abundant exits is used. Environmental information is represented with a static field and a dynamic one. Heterogeneity is achieved by authoring a set of static fields (called views) instead of a single one and individuals using different views. Communication happens only when another agent is occupying the cell current agent wants to move to. It consists of communicating agent sharing its view of the environment (static field it uses) and blocking agent changing its own accordingly.

FIPA is a standards organization operating under IEEE Computer Society. FIPA aims to produce software standards specifications for agent-based systems and MAS. It should be noted that, although philosophically related, the usage of term "agent" in AI and software communities differs. In the AI context, an agent is defined as something that acts but also has some attributes to distinguish it from a mere computer program such as autonomy, perception or adaptation capability [49]. On the other hand, in the software context, an agent is considered as a program that acts on behalf of its user [50]. In our work, we mostly adopt the general meaning in the AI context. However, FIPA organization focuses on developing standards for agent-based software. Therefore, to explain the ACL specification in this section, the term is generally used to refer to software agents.

Parameter	Description
performative	type of the communicative act
sender	agent sending the message
receiver	agent to receive the message
reply-to	agent that the replies should be sent to
content	content of the message
language	language of the content
encoding	encoding of the content expression
ontology	ontology(s) required to understand content
protocol	interaction protocol sender is employing
conversation-id	conversation identifier
reply-with	identifier that should be used in replies
in-reply-to	the message this one responds to
reply-by	time by which replies should be received

Table 2.1: Parameters in FIPA ACL Message Structure Specification [11]. The message structure in our approach uses the parameter subset indicated by the parameter names in italic. The message structure defines what is actually exchanged between the agents and it can be extended as needed.

ACL [11] specification is one of the most widely adopted standards of FIPA. It is a standard language for software agent communications similar to but superseding Knowledge Query and Manipulation Language (KQML) [51]. Both languages are based on the speech act theory by Searle [52].

FIPA ACL Message Structure Specification standardizes the message form. Table 2.1 shows the list of message parameters in the specification that can be extended by specific implementations according to the requirements of the application. The parameters that are included in a message are application dependent. The only mandatory parameter is *performative* but most messages are expected to also contain *sender*, *receiver*, and *content* parameters.

The performative parameter defines the type of the communicative act. A list of possible values for this parameter is suggested in the FIPA Communicative Act Library (CAL) Specification (see Table 2.2). The sender, receiver, and reply-to parameters can take values that stand for a participant in communication.

The content parameter is simply the content of the message. In some cases, such as a cancel message, the content is implicit and this parameter is not used. The language, encoding, and ontology parameters are all possible fields that can be used to describe the content when necessary. The remaining parameters, namely, protocol, conversation-id, reply-with, in-reply-to, and reply-by, are all about communication management.

The CAL list in Table 2.2 is given as an example. We do not directly adopt these communicative acts but instead build up a list of our own according to the needs of our applications. Yet naturally, our needs intersect with the CAL list and same or specific versions of these communicative acts such as failure, direction request, and inform (about path information), are employed. There are several additional FIPA ACL specifications each standardizing a different aspect of agent communication.

In summary, FIPA has various specifications that are aimed at standardizing software agent communications as a whole. Java Agent Development Framework (JADE) [53] is a well-known example framework to develop agent applications that are FIPA compliant. Although research on software and AI agents rarely meet at common points, our belief is that inter-agent communication in virtual crowds can become one such point.

2.4 Metrics

A metric we use in our evaluations is the *vfractal* estimation for movement trajectories. The term vfractal [12] refers to a collection of methods that estimate the fractal dimension [54] for animal movement trajectories. This estimation is a measure of the straightness/crookedness of the trajectories. Theoretically, the vfractal values range between one and two, one for a straight trajectory and two for a trajectory so tortuous that covers a plane. Biology-related literature uses vfractals commonly for animal movement paths. They have also been used to

Communicative act	Description
accept-proposal	accepting a previously submitted proposal
agree	agreeing to perform some action
cancel	sender no longer wants receiver to perform some action
cfp	calling for proposals to perform some action
confirm	informing the receiver that a proposition is true (receiver is uncertain about it)
disconfirm	informing the receiver that a proposition is false (receiver believes it to be true)
failure	informing the receiver that an attempted action failed
inform	informing the receiver that a proposition if true
inform-if	macro action informing the receiver either proposition is true or false (depending on what sender believes)
inform-ref	macro action informing the receiver about an objects reference to a descriptor
not-understood	informing the receiver that an action it took (e.g., a message it sent) was not understood
propogate	requesting the receiver to forward the embedded message (final receivers know the original sender)
propose	submitting a proposal
proxy	requesting the receiver to forward the embedded message (final receivers know the receiver)
query-if	asking whether or not a proposition is true
query-ref	asking for an object reference
refuse	refusing to perform an action
reject-proposal	rejecting a proposal to perform an action
request	requesting the receiver to perform an action
request-when	requesting the receiver to perform an action when a proposition becomes true
request-whenever	requesting the receiver to perform an action each time a proposition becomes true
subscribe	requesting to be informed whenever an object reference changes

Table 2.2: Communicative Act types in FIPA Communicative Act Library Specification.

evaluate agent-based simulation methods [55] and pedestrian evacuation behavior [56]. A trajectory is divided into pairs of fixed size steps for the estimation. Same values are calculated for randomly selected steps and they are averaged to obtain estimation results at that step size. Estimations are repeated similarly for varying step sizes. It is possible to calculate confidence values for the estimations made, which is an advantage of vfractal estimators.

Chapter 3

The Agent Architecture

In our final model [57], there are three major components to an agent: *perception*, *communication*, and *navigation* (see Figure 3.1). The main contribution of our work is the communication component in this architecture. Other parts are mainly what is required to apply the communication model in our example scenarios. Agents' need to move around is satisfied by the navigation component. The perception component provides services that are used both by the communication and navigation components. Each component contains subcomponents.

The perception component contains two further subcomponents, namely *hearing* and *sight*. A spherical volume around the agent is used to represent the hearing range and a pyramid shape in front of the agent represents sight volume (cf. Figure 3.2). Important objects such as other agents, doors, or signs that are in sight and/or hearing range are continuously tracked by these subcomponents. The primary task of these subcomponents is to provide this tracking data to other components when requested. The arrows from the perception subcomponents to the other components in Figure 3.1 shows this relationship.

Both perception subcomponents function via collision detection methods. Every object of interest in the scene, such as a door or an agent, has an attached invisible convex volume, called a *collider*, which encloses the object and is as

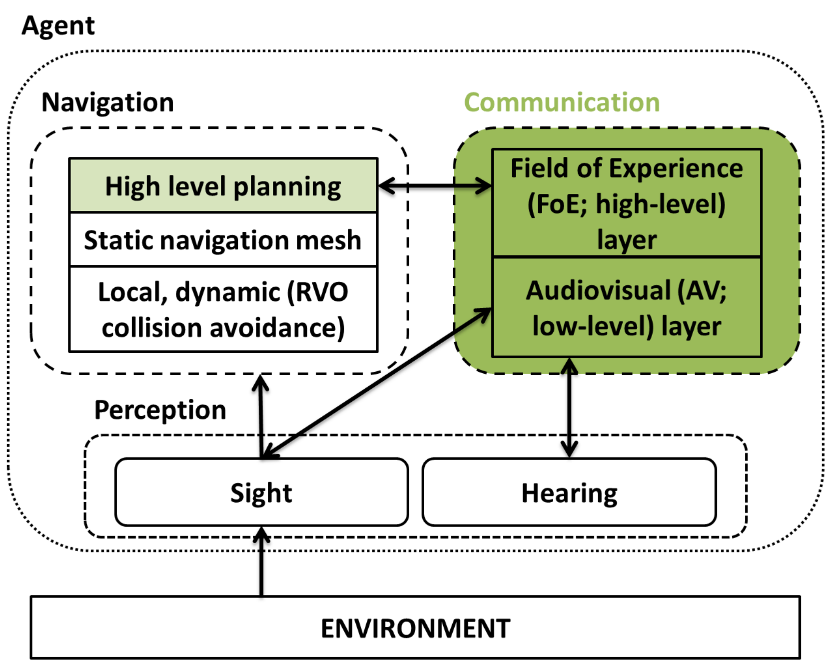


Figure 3.1: The agent architecture with its three major components: communication, navigation, and perception. The internal structure of components and their relationships are also shown. The subcomponents of perception, namely sight and hearing, track objects (e.g., doors) and other agents in sight and hearing range respectively. Navigation consists of three layers: a local collision avoidance solution (Reciprocal Velocity Obstacles; RVO), a global path planning solution (navigation mesh), and a higher-level, scenario-dependent planning part. Planning can be as simple as deciding a single target position at the beginning or it can be a complex algorithm to simulate decision-making during evacuation (cf. Figure 6.3). Navigation does not require the communication component but cooperates with it when it is enabled. Communication separates message/scenario-dependent (high-level, FoE) tasks from those that are message/scenario independent (low-level, Audiovisual, cf. Figures 4.2 and 4.3). The green color indicates the novel components of our approach.

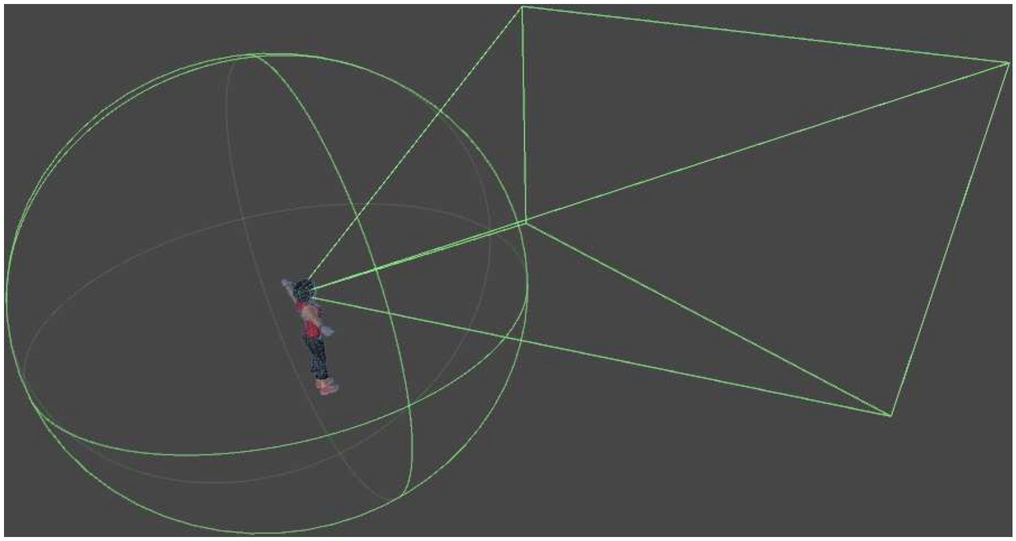


Figure 3.2: The visualization of hearing and sight volumes.

close to its visible shape as possible. The volumes in Figure 3.2 representing the perception subcomponents are in fact also colliders. Collisions of these colliders are constantly checked by the game engine's background mechanisms. We define the tasks to be carried out when an object's collider collides with the perception collider and when an existing such collision ends. These tasks are simple list management tasks. Both hearing and sight subcomponents keep lists of objects of interest that lie within the representing volume.

The perception component does not have strong dependencies with the communication component or navigation component. It provides services (lists of objects of interest) that are used by the other components as needed. A major and constant consideration during design and development has been the fact that the current perception implementation could be changed, replaced, or extended. For example, a user of our communication model might want to combine it with a different hearing model or to include a smell capability for agents. In such cases, the required modifications in navigation and communication components will be minimal. Additionally, these modifications will almost always take place inside the higher level subcomponents.

The navigation (i.e., path planning) problem, in general, consists of reaching a given goal position from an initial position. The environment may contain only stationary obstacles in which case, it is called a static environment or, it may be dynamic, meaning that it contains moving obstacles. In a dynamic environment, navigation involves both calculating a collision-free path to the goal position and following this path successfully without collisions with moving obstacles.

It is a common solution to consider navigation in a dynamic environment in two stages as global and local navigation. The global navigation considers only the static obstacles in the environment. Its task is to calculate a path to the goal avoiding the static obstacles. Graph-based roadmap methods are popular solutions to this problem [58]. Directly following the path calculated by the global navigation solution is likely to result in a collision with a moving obstacle. Therefore, the local navigation stage applies collision prediction and avoidance techniques. The path is followed while nearby dynamic obstacles are taken into account and avoided.

Our navigation component splits the task similarly and makes use of well-known existing methods. The global navigation, which is the middle subcomponent in the navigation component in Figure 3.1, uses a precomputed static navigation mesh [59] to calculate static obstacle avoiding paths. The navigation mesh is a mesh of convex polygons and defines the navigable areas of the environment. This structure can also be considered as a graph with each polygon as a node. Two nodes are connected if the corresponding polygons are adjacent. A path can be calculated by graph search methods such as A* [60]. The static navigation meshes for our example environments are generated automatically from the scene geometries by specifying some parameters related to agent shape and movement such as agent and maximum step heights.

The bottommost, local navigation subcomponent uses Reciprocal Velocity Obstacles (RVO) [18] for avoiding dynamic obstacles (i.e., other agents). RVO extends the Velocity Obstacles (VO) method [61], which was developed for a robot in a dynamic environment where dynamic obstacles are passive. When the dynamic obstacles are other agents that are also avoiding collisions, VO has some drawbacks. In order to overcome these drawbacks, RVO assumes that the dynamic obstacle is active, i.e., it is capable of similar collision avoidance reasoning.

A third navigation layer at the top in Figure 3.1, called the planning layer, controls the use of this two-layer navigation capability. This planning layer is scenario dependent. In some scenarios, planning is simple and straightforward such as only specifying a final destination. In other scenarios, on the other hand, it can involve more computation. For example, intermediate targets are used instead of calculating a path from the current location to the destination when we want to simulate exploration in the case that an agent has no knowledge about the building.

Chapter 4

The Communication Model

We focus on deliberate (i.e., intentional) communication. This boundary is necessary because it can easily become difficult to decide whether an interaction is a communication or not. The definition of communication according to the Oxford dictionary [62] is:

> "the imparting or exchanging of information by speaking, writing, or using some other medium."

This definition appears to be clear at first; however, it is easy to get confused when one begins to consider what it includes. Let us consider a hypothetical scenario. Two people, A and B, are situated in a room and A watches as B stands up and turns towards the door. During this, at some point in time, A will understand that B is going to exit, however, this new information was not specifically exchanged. It could be said that there is no communication in this example. But, what if it was actually B's intention that A will understand the situation? With this consideration, it becomes possible to say that there is communication here as B is sending a message to A by her/his actions. This simple scenario tells us that whether an interaction in real-world is communication or not may be difficult to answer and that real-world communication is a complex

Layer	Function
7. Application	Contains variety of protocols commonly needed by users
6. Presentation	Data representation, encryption and decryption, converts machine dependent data to machine independent data
5. Session	Inter-host communication and session management between applications
4. Transport	End-to-end connections, reliability and flow control
3. Network	Path determination, logical addressing, and routing
2. Data link	Transforms a raw transmission facility into an error free line
1. Physical	Transmitting raw bits over a communication channel

Table 4.1: Open Systems Interconnection communication model layers.

concept. Moreover, we have experienced that descriptions such as direct/indirect or implicit/explicit communication are not well-defined, i.e., they are sometimes used to refer to different concepts. We do not have the expertise and are not attempting to provide a better definition for communication. However, because of the problems described here, for our purposes, we limit ourselves by only considering deliberate communication. It should be noted that when we say communication or information, we are not referring to a particular form such as speech. We are considering these concepts at an abstract level as communicative intentions and meanings in our minds.

Layered network protocols (e.g., Open Systems Interconnection (OSI) and Transmission Control Protocol/Internet Protocol (TCP/IP)) used for computer networks can be considered as successful applications of transmission models of communication. These layered architectures have become world wide standards for their well-known advantages in implementation [63]. First of all, layers reduce the complexity of the design. Each layer provides services to the higher layers while hiding the implementation details of those services. Well-defined interfaces between the layers make it simpler to modify or completely replace a layer's implementation. When we consider an agent in a virtual environment, a layered architecture can also be useful. Let us consider the layers of OSI model shown in Table 4.1.

Each layer in OSI (as in other communications system models) provides services to the layer above and uses the services of the layer below. Commonly, Layers 1-3 are called media layers and Layers 4-7 are called host layers. In our context of an agent in a virtual environment, we can match this distinction with the embodiment and the mind of the agent. A layered architecture will have similar advantages in this context. Cognitive (mind) layers can be responsible for cognitive skills such as interpretation of messages or choosing how to communicate a particular message. On the other hand, embodiment layers can represent body related issues such as perceiving or actuating of signals.

We presented our communication model in its early stages at The 27th Conference on Computer Animation and Social Agents (CASA'14) [64]. The initial design at the time contained three layers. Our aim at these early stages was to make the communication model as human-like as possible and to use it with a more complex agent architecture. It was realized that the human-like complexities, such as agent personality affecting communication capability, were not clearly observable when viewing a crowd. Therefore, instead of making the model as human-like possible and using a complex agent architecture that includes personality, we started working with a simplified agent architecture and targeting a generic communication model. By generic, we mean a model that can be used in new scenarios and combined with different agent architectures with as little effort as possible. Currently, we have a two-layer communication model. Compared to the aforementioned paper, currently, a more realistic simulation environment and improved (3D) visualization are used and a more formal evaluation is provided.

When considering the actual layers within embodiment and cognitive categories, our aim is not to maintain a one-to-one similarity with OSI but to keep the architecture as simple as possible and only introduce layers that we think are necessary. Table 4.2 shows the initial three-layer architecture we employed. The middle N-log layer was later dropped to reach the current two-layer design.

Out of these three layers, only the lowest, Audiovisual (AV) one was thought as an embodiment layer. The other two layers were considered as cognitive layers. We tried to keep the number of layers as low as possible (compared to a model like

Layer	Function
Field of Experience (FoE)	Encoding and decoding with respect to the field of experience
N-log	*Controls dialogs (or multilogs) between agents*
Audiovisual (AV)	Perceiving or producing of auditory or visual signals

Table 4.2: The initial three-layer architecture of our communication model. The N-log layer is in italics because it was later dropped to obtain the current two-layer design.

OSI). For example, we did not include any layers corresponding to the Network and Data Link layers of the OSI. The main reason for the first one is that when we, as humans, receive a message to be routed, we go through the interpretation process before routing. Hence, we believe that routing tasks, when they occur, should be handled by the decision making processes. The reason for the second one (not having a layer corresponding to Data Link layer of OSI) is that including phenomena such as unclear messages that include errors or missing parts due to noise in the environment will further complicate the implementation attempt. As a result, we ignored such phenomena and did not need error detection and correction facilities. Nevertheless, the model can easily be extended with another layer for this purpose. We want to emphasize that the layers we include are not taken as complete or final. It is possible to easily extend the model.

The N-log layer was for slightly higher level tasks such as maintenance of dialogs (or multilogs) from their beginning to finalization, which can span short/long durations. Its aim was to control initiation, continuation, and finalization of dialog (or multilog) instances. However, the example applications of the model requires only a few simple dialogs. Therefore, this layer was removed and its responsibilities were handled by the FoE layer in our examples. If the model is to be used in new scenarios and different types of or flexible dialogs are needed, N-log layer can be reintroduced. Its implementation allows agents to establish sessions between them. This involves dialog control, i.e., keeping track of whose turn it is to send a message. The reason we called it *N-log*, not *Dialog*, is that communications among more than two agents may also be needed. However, it is much easier to implement facilities for dialogs. One way to handle multilogs is to treat them as multiple dialogs. Whether this treatment is satisfactory or issues regarding its limitations are not topics we engaged in our work.

With the removal of the N-log layer, the communication process is only simplified by a two-layer design. The two layers separate high-level and low-level tasks. The high-level tasks are those that depend on what the message is. On the other hand, the low-level tasks are independent of the message type. We named the upper layer after the FoE concept in Schramm's communication model as FoE layer and the lower layer is called the AV layer. Responding to a direction request is an example FoE task. On the other hand, turning towards the receiver when sending a message, moving closer if (s)he is not close enough, signaling from a distance to catch attention are examples of AV tasks.

When considering real communication, it is more accurate to consider signal creation, transmission, and reception as the lowest-level tasks. However, as we mentioned before, we take communication at an abstract level, and hence, signal creation and reception phenomena are not simulated. Two AV tasks, namely `transmit` and `receive`, simply fill in for these signal-related concepts. We assume that when an agent's `transmit` routine is called, the given message is sent, and similarly, when an agent's `receive` routine is called, the message is perceived. Through this abstraction, same mechanisms should work for different forms of communication such as speech, writing, signaling, and others.

Having AV and FoE layers separates the high-level communication intentions from the low-level mechanisms making communication possible. The communication model becomes more generic through this separation. In other words, the layered design makes it easy to apply the communication model in new scenarios and to combine it with different agent architectures. When new message types are needed in a new scenario, the modifications will be local to the FoE layer. Similarly, for example, if the agent architecture is to include a different perception capability, the required changes will mostly be in the AV layer.

We also sometimes use the terms *meaningful* and *hollow* communications. These terms allow us to treat communication instances that have an influence on behavior as a special case. Meaningful communication is used to refer to the type of information exchange that can influence behavior. For instance, agents communicating about exit routes in an evacuation scenario is meaningful communication. Hollow communication refers to the type of communication where there is no exchange of important information that affects agent behavior. For example, in a scenario with pedestrians on a street, agents may talk with each other without any information exchange to affect behavior. In both meaningful and hollow communication instances, same low-level routines can be used as a result of having separate FoE and AV layers. Only in the FoE layer, the difference between the two types of communication will be apparent.

The complex notion of communication or a specific form of communication such as a language becomes easier to deal with when we consider the syntax, semantics, and pragmatics for that notion. Developers of ACL and similar languages mostly specify the syntax and semantics for their language but they also employ some ideas observed in natural language pragmatics. Specification of a common and extensible syntax is often the first task. In our model, the syntax is defined by the message structure. Semantics is more about the underlying meaning. Our message types are the core elements of the semantic specification. Pragmatics is about the relationship between the context and meaning. The fundamental pragmatics elements in our model are the high-level (message/scenario dependent) protocols used.

The form of exchanged information, i.e., the syntax, is specified by the message structure. We partially adopted the ACL (see Table 2.1) parameters of FIPA according to our needs. What we call message types in our model is the same as the performative parameter options in FIPA ACL. Moreover, `source`, `destination`, and `content` parameters are also made use of. The types of messages we used in our example scenarios were created as necessary. The present collection of applied message types, which corresponds to the semantics in our model, are:

- WAVE: the type of message that is used when the sender wants to attract the receiver's attention; this message type will be necessary when the sender is in receiver's sight but not in hearing range;

- CHAT: the type of message that is used to simulate hollow communication;
- DIRECTION_REQ: the message type that is used in asking about direction; a target location can be included in the message content;
- PATH: a message type that is used in responding to a DIRECTION_REQ; message content includes a full path;
- FINAL_AND_NEAR_TARGET: a second possible response to a DIRECTION_REQ; message content only includes a final and a nearby intermediate target location;
- EXIT_THROUGH: a third possible response to a DIRECTION_REQ; message content includes a particular navigable object such as a door or staircase;
- FAILURE: the type of message that is used when a meaningful response cannot be given.

When our communication model is to be used in a new application, new message types can be added to this collection as needed. One of the advantages of the layered architecture in our application is that the highest level communication intentions are abstracted away from the modules facilitating the communication between agents. Our agents will very likely have different reasons for communicating in different scenarios. For example, in a building evacuation scenario, agents need to communicate information regarding exit routes. But in a different scenario, this particular communication intention may be inessential and others may be needed. The layered architecture allows us to treat these highest level intentions in a similar way with communication requests of different applications on a computer system. A file transfer application sends and receives very different messages compared to a video teleconferencing application, but they make use of same lower levels in the TCP/IP protocol. In short, in the proposed layered architecture, it is easy to introduce new message types at the FoE layer as they are needed by new scenarios.

4.1 Audiovisual (AV) Layer

This layer represents the low-level sending and reception capabilities that are independent of the message type. An agent sending a message needs to realize a speech act or gesture. Also, each agent is required to be aware of its surroundings in order to receive messages. This layer mainly deals with simulation of these phenomena, therefore, it relates to the perception components of the agent architecture and the AV layers of other agents.

The main responsibilities of the AV layer are to deal with transmitting a message passed from the upper layer to the destination and to handle an incoming message. However, simulating a human-like perception and expression capability for agents (the major issue driving the field of ECAs) is not a contribution of our thesis. Speech recognition or synthesis are not topics we deal with. Transmissions in both directions are essentially implemented with simple method calls and animation of the virtual character. Here, the animation is only for viewer's visual understanding of the act and not taken as a communicative act by other agents. Other agents only consider the message sent via the method call.

If the communication is directed to a specific agent, then the AV layer tries to locate that agent. Whether the message is sent directly or not depends on the position of the sender with respect to the receiver and the orientation of the receiver. We assume a spherical area around an agent for hearing and a pyramid shaped area in front of the agent for seeing as was shown in Figure 3.2. A top-down view of this perception model is given in Figure 4.1. The diagram shows the perception areas of a receiver that is at the center of the circle.

A message to be sent can be a gesture such as waving or a spoken question. If a sender is both inside the hearing range and sight of a receiver (i.e., in Area I), then message sending can directly take place no matter what the message is. If the sender is in Area II then speech is the only option, while gesture is the only option if the sender is in Area III. In each of these cases, to continue the communication after the initial message, agents may need to move or the receiver

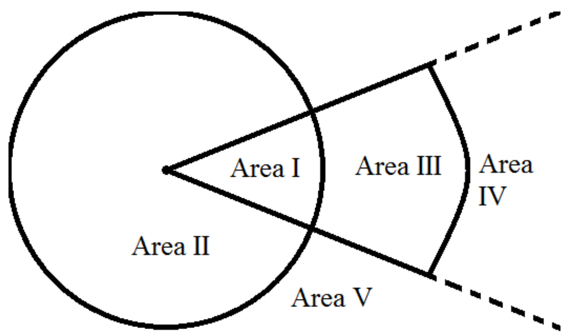


Figure 4.1: A top-down view of perception areas. The receiver agent is at the center of the circle. The circle represents hearing range and the conical area in front represents sight range.

agent may need to turn towards the sender changing its orientation. If the sender agent is in Area IV or V according to the receiver, then it has to move towards other areas in order to send the message. When the sender is in a location that makes gesturing or speaking possible, then the `receive` method of the receiving agent is called. It is also possible that a message is passed down to the AV layer to be broadcast. In this case, the `receive` methods of all nearby agents are called. Animations are invoked properly together with these method calls so that communication behavior is visible to an observer of the simulation.

The flowchart in Figure 4.2 summarizes the implementation for low-level message sending tasks in the AV layer. Assume that when an agent wants to send a spoken message to another agent, we are at the `Start` node. The sender first checks whether the receiver is in sight. The agent needs to explore its surroundings if the answer is no (i.e., when the receiver is not in sight). Otherwise, the sender should determine whether the receiver can hear her/him or not. The spoken message can be sent when the sender is within the receiver's hearing range. When the sender is outside the hearing range, (s)he needs to move towards the receiver. There are two possibilities in this situation. If the sender is in sight of the receiver despite being outside the hearing range, then (s)he can Wave to attract the receiver's attention and then move. The Wave message causes the receiver to stop if (s)he is also moving and prevents a possible communication failure. In the second possibility, the sender is both outside the hearing and sight ranges and the only thing (s)he can do is to move towards the receiver.

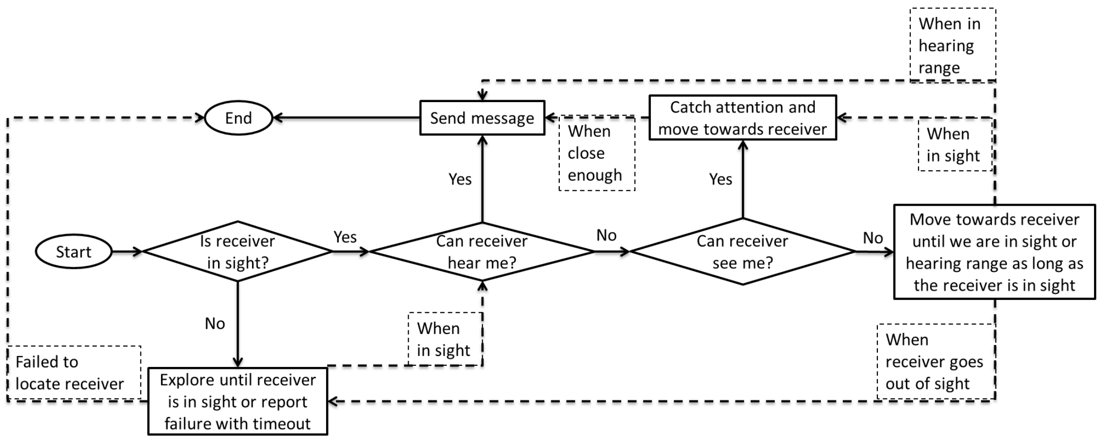


Figure 4.2: The flowchart for the Audiovisual layer message sending procedure. When there is a spoken message to be sent, the sender agent begins from the `Start` node. Message/scenario type is irrelevant. There two possible results after following this routine. The message is sent, or the agent failed to do so.

Our communication component is implemented as a push-down automaton (PDA) [65]. There are two reasons for this design decision. The first reason is that it is natural and straightforward to represent an agent's communicative situation at any point as a communication state. The second reason is that in some situations, an agent needs to backtrack to the previous state. For example, after attracting attention, a sender agent needs to roll back so that it can send the original message. PDA's stack structure facilitates recording state changes; as a result, it enables rolling back when needed. The PDA states for the AV layer, together with the mathematical abbreviations we use, are:

- Not_Communicating (q_{NC}),
- Wants_to_Communicate (q_{WtC}),
- Found_Comm_Target (q_{FCT}),
- In_Hearing_Range_of_Target (q_{IHRoT}),
- Waiting_for_Transmission (q_{WfT}),
- Waiting_for_Transmission_Moving_Closer (q_{WfTMC}),
- In_Sight_of_Target (q_{ISoT}),

- MOVING_CLOSER (q_{MC}),
- NOT_IN_SIGHT_OR_HEARING_RANGE (q_{NiSoHR}),
- RECEIVED_A_MESSAGE (q_{RaM}), and
- ATTENTION_CAUGHT (q_{AC}).

Most state names themselves explain the states' purposes. There are two states that are used when the receiver of a message is busy: WAITING_FOR_TRANSMISSION and WAITING_FOR_TRANSMISSION_MOVING_CLOSER. The state diagram for the AV layer part is shown in Figure 4.3. *Any State* is a placeholder to represent all other states. Each transition to a state S from *Any State* actually stands for the set of transitions to S from every state. For instance, the transition to ATTENTION_CAUGHT from *Any State* allows an agent to successfully receive a WAVE message no matter what state (s)he is in. The transition conditions are not given in the figure for clarity. They are often natural conditions expected for that state change. For example, the transition to IN_SIGHT_OF_TARGET from FOUND_COMM_TARGET state happens with the condition that this agent (the sender) is in sight of the receiver agent that was found previously.

Formally, a PDA M is traditionally defined as a septuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where Q is a finite set of states, Σ is a finite set of input symbols, Γ is a finite set of stack symbols, δ is the transition function with $\delta : (Q \times \Sigma \times \Gamma) \rightarrow (Q \times \Gamma^*)$, q_0 is the starting state, Z_0 is the initial stack symbol, and $F \subseteq Q$ is the set of accepting states. We use the $*$ symbol to indicate the Kleene closure for a set. The PDA that we use has some differences. State changes are triggered by certain conditions rather than the user input. Therefore, it is more appropriate to call Σ as a condition set but mathematically there is no difference. The other difference is that our PDA is not intended to recognize strings or accept certain situations but rather to control agent actions. Hence, instead of accepting states (F), we need a set of actions A and an action function, $\alpha : Q \rightarrow A^*$. As a result, our PDA M' can formally be defined as an octuple $M' = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, A, \alpha)$. The elements are:

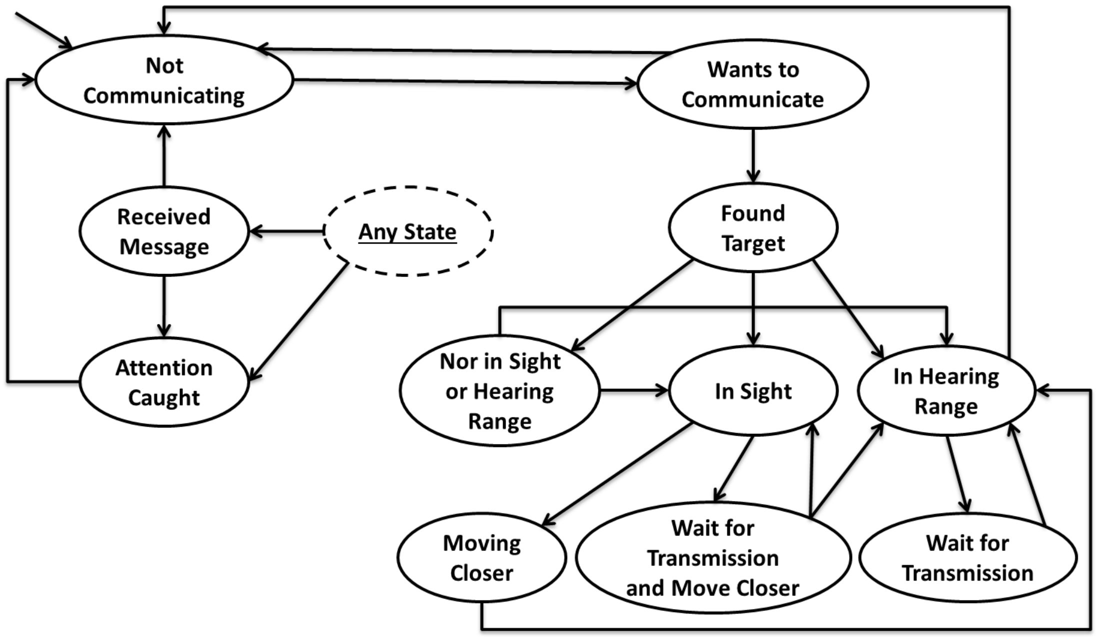


Figure 4.3: The Audiovisual layer state diagram. Transition conditions are omitted for clarity. Any State is a placeholder representing all other states. Towards the left, we have the states that are mostly related to message reception, while on the right, we have those that are related to message sending (cf. Figure 4.2).

- Q, the state set, is a union of the AV layer states $Q_{AV} = \{q_{\mathrm{NC}}, q_{\mathrm{WtC}}, \ldots, q_{\mathrm{AC}}\}$ listed above and the FoE layer states (cf. next section), i.e., $Q = Q_{AV} \cup Q_{FoE}$;
- Σ, the finite set of conditions, contains boolean values representing various situations such as another agent being in sight or the target of communication being busy, $\Sigma = \{$`CommunicationNeed`, `SomeoneInSight`, `TargetCanHear`, `TargetCanSee`, `TargetBusy`, `MessageSent`, `IncomingMessage`, `WavedAt`, `MessageProcessed`$\}$;
- Γ, the stack symbols, are same as the states, i.e., $\Gamma = Q$;
- δ, the transition function, is given in Table 4.3;
- $q_0 = q_{\mathrm{NC}}$ is the starting state;
- $Z_0 = q_{\mathrm{NC}}$ is the initial stack symbol;

	`CommunicationNeed`	`SomeoneInSight`	`TargetCanHear`	`TargetCanSee`	`TargetBusy`	`MessageSent`	`IncomingMessage`	`WavedAt`	`MessageProcessed`
q_{NC}	$q_{\text{WtC}} \downarrow$	-	-	-	-	-	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{WtC}	-	$q_{\text{FCT}} \uparrow\downarrow$	-	-	-	-	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{FCT}	-	-	$q_{\text{IHRoT}} \uparrow\downarrow$	$q_{\text{ISoT}} \uparrow\downarrow$	-	-	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{IHRoT}	-	-	-	-	$q_{\text{WfT}} \uparrow\downarrow$	$q_{\text{NC}} \uparrow$	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{WfT}	-	-	-	-	-	$q_{\text{NC}} \uparrow$	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{WfTMC}	-	-	$q_{\text{WfT}} \uparrow\downarrow$	-	-	$q_{\text{NC}} \uparrow$	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{ISoT}	-	-	$q_{\text{IHRoT}} \uparrow\downarrow$	-	$q_{\text{WfTMC}} \uparrow\downarrow$	-	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{MC}	-	-	$q_{\text{IHRoT}} \uparrow\downarrow$	$q_{\text{ISoT}} \uparrow\downarrow$	-	-	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{NiSoHR}	-	$q_{\text{FCT}} \uparrow\downarrow$	$q_{\text{IHRoT}} \uparrow\downarrow$	$q_{\text{ISoT}} \uparrow\downarrow$	-	-	$q_{\text{RaM}} \downarrow$	$q_{\text{AC}} \downarrow$	-
q_{RaM}	-	-	-	-	-	-	-	$q_{\text{AC}} \downarrow$	$q_{\text{NC}} \uparrow$
q_{AC}	-	-	-	-	-	-	$q_{\text{RaM}} \downarrow$	-	-

Table 4.3: The transition function (δ) of the push-down automaton. The rows correspond to different states and the columns correspond to different conditions. Each table entry gives the next state and the stack operation(s) for a particular state transition. An up arrow ($\uparrow$) indicates a stack `pop()` operation and a down arrow ($\downarrow$) indicates a stack `push(name)` operation where `name` is the name of the new state.

- A = {`stop()`, `turnToTarget()`, `transmit()`, `wave()`, `moveCloser()`, `wait()`, `turnToSource()`, `resume()`} is the set of actions; and
- $\alpha : Q \rightarrow A^*$, the action function, is given in Table 4.4.

4.2 Field of Experience (FoE) Layer

Going up the hierarchy, the FoE layer represents individual encoding and decoding tasks. In Schramm's communication model (cf. Section 2.2), encoding and decoding happens with respect to a communicator's FoE (its knowledge, experiences, beliefs, and so on). It is the responsibility of the FoE layer to turn a communicative intent into a message (or vice versa) according to the state of the agent. The message and/or scenario dependent protocols used in this layer specify the pragmatics.

State	Action
q_{NC}	`resume()`
q_{WtC}	-
q_{FCT}	`stop()`, `turnToTarget()`
q_{IHRoT}	`transmit()`
q_{WfT}	`wait()`
q_{WfTMC}	`moveCloser()`, `wait()`
q_{ISoT}	`wave()`, `moveCloser()`
q_{MC}	`moveCloser()`
q_{NiSoHR}	`moveCloser()`
q_{RaM}	`stop()`, `turnToSource()`
q_{AC}	`stop()`, `turnToSource()`

Table 4.4: The action function (α) of the push-down automaton

There can be various communicative intentions among many different scenarios. This layer will evolve as new scenarios are developed and new information that agents can share is introduced. There is an analogy here with new protocols being required and developed in computer networks as new applications making use of the network emerge. For the time being, we only consider the fact that the data to be communicated will be coming from (or going to) the components of the agent state or the decision making processes. Therefore, one key issue is about recognizing what the message is about.

The PDA described in the previous section is actually extended by the FoE layer. There will be additional communication states, transitions, and actions needed in an application scenario by the FoE layer tasks. FoE layer states were not included in the description for brevity and generality. In our application scenario examples, we used two FoE layer states. These are DIRECTION_REQUESTED and IN_CHAT states. It should be obvious that these states are scenario and message type dependent.

Chapter 5

Preliminary Experiments and a Change of Direction

The initial layered design was different from our final model. It was meant to be combined with a more complex agent architecture. The agent attributes considered at the time are given in Table 5.1. The static psychological attributes are collectively known as the OCEAN model of personality or the Big Five Personality Traits. They have been successfully applied in crowd simulations to achieve agent heterogeneity [6]. The rest of the attributes can be easily understood from their names. The static physiological agent attributes represent physical capabilities of an agent such as how fast it can move, stop, or turn and how far it can see or hear. The dynamic attributes represent parts of the agent state which can change during a simulation. Preferred speed is the magnitude of the velocity that the agent prefers to move with when there are no constraints on movement, i.e., when there are no imminent collisions. Stress and fatigue represent the mental and the physical tiredness of the agent respectively. “Is Injured” attribute is a boolean value to simulate a major deterioration of agent capabilities.

Various preliminary experiments were carried out in order to test the influence of such agent attributes on communication. These are grouped under three different experiment sets explained in sections below. We concentrated on varying

	Psychology	Physiology
Static	Openness Conscientiousness Extraversion Agreeableness Neuroticism	Maximum Speed Maximum Acceleration Minimum Acceleration Maximum Angular Speed Hearing Range Sight Range
Dynamic	Stress	Preferred Speed Fatigue Is Injured

Table 5.1: Agent attributes for the earlier agent design

only a small subset of attributes among agents so that the effects can be noticed more clearly. The static physical attributes are kept the same for all agents and stress is not simulated. Depending on the particular set and a particular scenario in that set, the static psychological attributes and/or fatigue are varied or not.

5.1 Experiment Set I

Three experimental simulation scenarios formed the main discussion of our first paper [64] during the early stages of our work. A single floor building with two exits was used in an evacuation scenario. None of the agents knew the building at start. The connectivity of the building parts were represented with a CPG [5]. Each agent has its own CPG to represent its knowledge of the building. This personal copy can be any subgraph of the full building CPG. In the experiment, an agent's personal CPG, initially only includes the cell (room) the agent is in and the cells that are one-adjacent to that cell. The evacuation behavior is summarized in the flowchart in Figure 5.1. Here, the key point connecting agent attributes and communication (initiation) is the probability calculation. The more extrovert and social an agent is, the higher the probability, and hence, the more likely the agent is to ask for directions.

Another probability is calculated when an agent receives a message. Again, personality values influence this probability. At this point, receiving agent decides

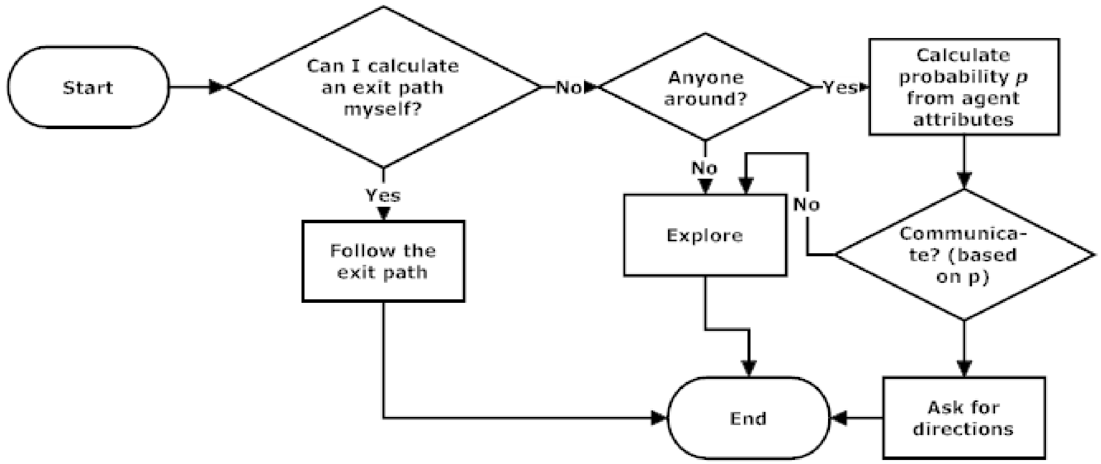
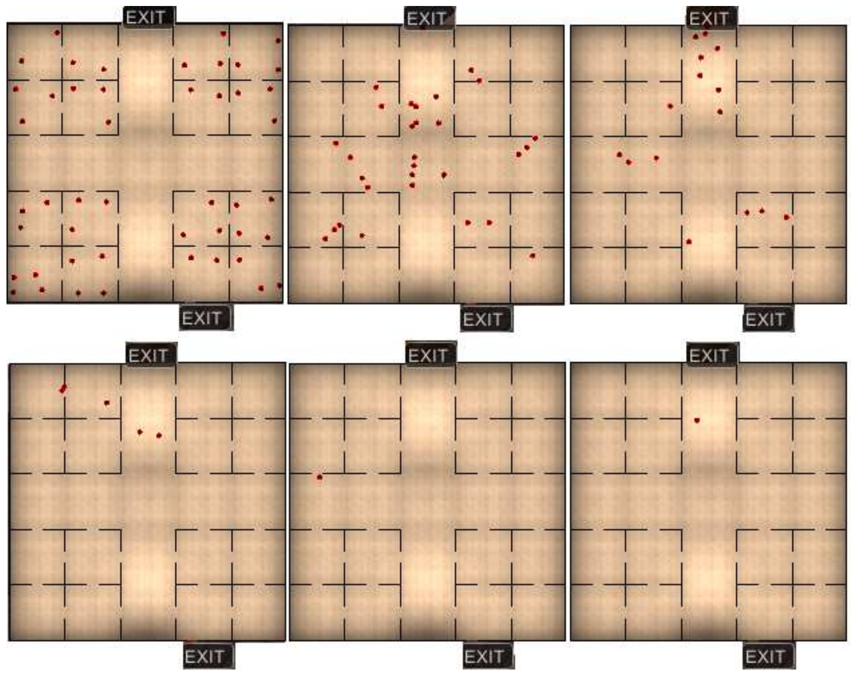


Figure 5.1: The flowchart describing the evacuation behavior. This is an earlier and simpler version of the high-level planning algorithm we developed for an agent in an unknown environment (see Figure 6.3).

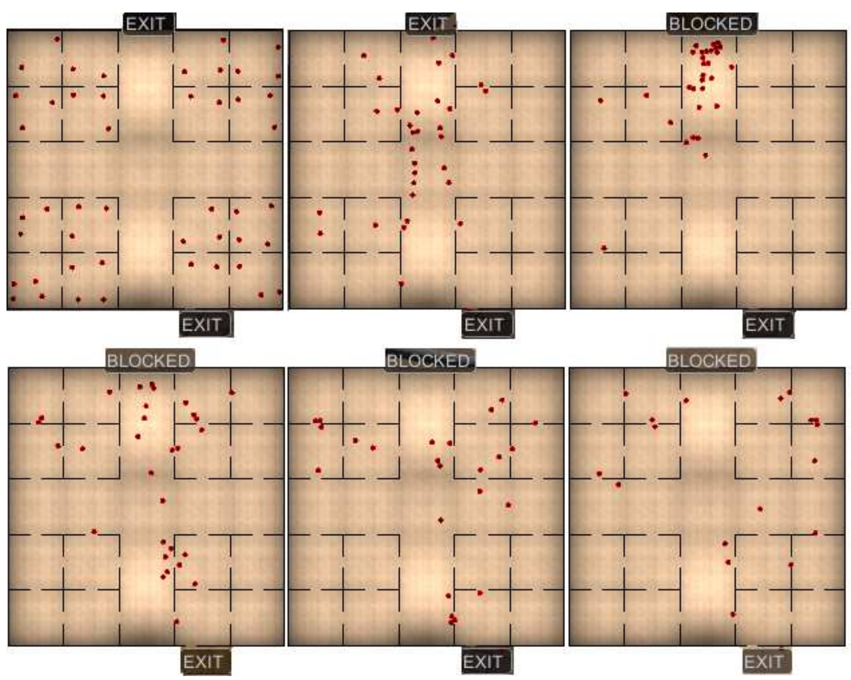
whether it will evaluate and respond to the message properly. The more introvert the agent is, the more likely it is to directly respond in a negative way and cut short the attempted dialog. Here, we observe a case in which agent attribute affects communication progression (and hence, conclusion and outcome).

In the first of three simulation scenarios, agent personalities are not varied, i.e., all agents have the same, medium (in terms of introversion/extraversion) personality parameters. The simulation is carried out until all agents evacuate the building with no specific events occurring (cf. Figure 5.2a). It is not quite possible to demonstrate the communications taking place over the course of the simulation with few images. Therefore, in Figures 5.2a and 5.2b, we provided a top-down view of the simulation to demonstrate path planning and evacuation behaviors.

To be able to show the effect of fully autonomous communication capabilities of agents, the second simulation scenario involves a hazardous event. The main door of the building malfunctions 20 seconds into the simulation so that it cannot be used by agents afterwards. Everything else is the same as the first simulation. It is observed that the agents decide completely autonomously to recalculate their paths and share information with others (cf. Figure 5.2b).



(a) Scenario 1



(b) Scenario 2

Figure 5.2: Still frames (frames 1, 91, 181, 271, 361 and 451 left to right, top to bottom) from the simulation. Agents have the same average personality. (a) No events. (b) The main door becomes unusable after 20 seconds.

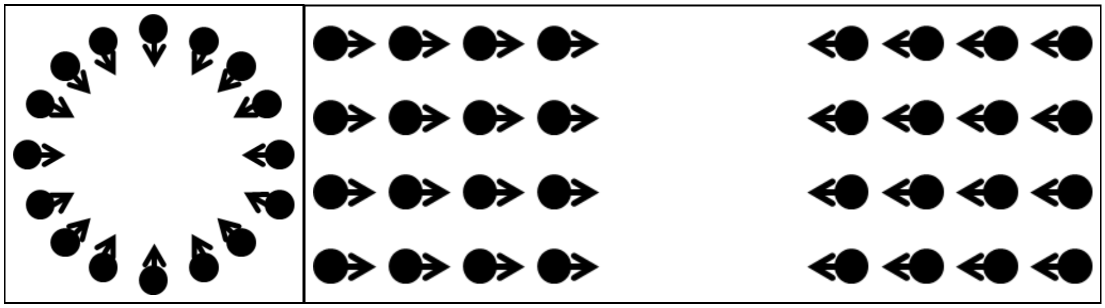
Finally, in order to show the effect of personality to communication, the last simulation scenario involves varying personalities for agents. The female characters are assigned extrovert personalities (0.9 in the scale $[-1, 1]$) whereas male characters are assigned introvert personalities (-0.9 in the scale $[-1, 1]$). Everything else is the same as the second simulation scenario. A figure is not provided for this case as it will be very similar to the previous one and as it will not demonstrate much.

5.2 Experiment Set II

A second group of experiments involved agents starting in a specific formation in an open environment without any obstacles. The only type of communication used is chatting, which does not involve any information passing (i.e., hollow communication). Our aim here was to evaluate attributes' effect only on communication initiation. All communications were standard in terms of duration. We were interested in the mean number of interactions ($\bar{x_i}$) for all agents in a simulation run and the standard deviation (σ) for this distribution.

There are two different scenarios in this experiment set that differ in the initial and final formations of the crowd (see Figure 5.3). In Scenario 1 experiments, the agents are placed regularly on the circumference of a circle. In these setups, each agent's goal position is the antipodal position (i.e., the opposite of start position with respect to the center of the circle). In other words, the ideal path for each agent is the diameter from the starting point. All ideal paths meet at the center of the circle. In Scenario 2 experiments, the agents are regularly placed in two opposing rectangular areas and the goal position for an agent is given as the same position within the opposite rectangle.

Experiments in both scenarios are carried out for differing number of agents. Scenario 1 is simulated for 24, 60, and 120 agents, whereas Scenario 2 is simulated for 50, 72, 98, and 128 agents. For each of these cases, there are two settings: one in which all agents have standard personality and personality has



(a) Scenario 1 (b) Scenario 2

Figure 5.3: The two scenario formations used for agent placement and goal position assignment.

# agents	24		60		120	
	$\neg P$	P	$\neg P$	P	$\neg P$	P
$\bar{x}_i$	1.75	1.75	1.3	1.33	1.56	1.7
σ	0.53	0.68	0.53	0.57	0.58	0.66

Table 5.2: Scenario 1 experiment results for mean number of communications ($\bar{x}_i$) and standard deviation (σ).

.

no effect on communication ($\neg P$) and the other one (P) with varying personalities and personality affecting communication initiation probability. The results are summarized in Tables 5.2 and 5.3.

# agents	50		72		98		128	
	$\neg P$	P	$\neg P$	P	$\neg P$	P	$\neg P$	P
$\bar{x}_i$	0.68	0.68	0.61	0.53	0.69	0.63	0.55	0.40
σ	0.68	0.82	0.52	0.56	0.56	0.62	0.54	0.55

Table 5.3: Scenario 2 experiment results for mean number of communications ($\bar{x}_i$) and standard deviation (σ).

.

5.3 Experiment Set III

In the third set of experiments, our aim is to simulate a possible effect of dynamic agent attributes. The communication routine is modified so that a single dynamic attribute, namely fatigue, could alter communication initiation probability. Fatigue represents the physical tiredness for an agent and its effect is that the more tired an agent is, the less likely it is to communicate.

Each agent is assigned a random fatigue value in the range $[0, 1]$ at the beginning of the simulation. During the simulation, fatigue increases at a standard rate (an exaggerated rate of approximately 0.0017 units per frame is used below). In order to observe the outcome, active number of conversations are counted during a simulation with five second intervals. We used the same start and stop formations as Scenario 1 of Experiment Set II. Simulations are executed with three different settings for agents.

1. *NoPersNoFtg*: Agent personality and fatigue has no effect on communication initiation.
2. *PersNoFtg*: Agent personality affects communication initiation, but fatigue does not.
3. *PersFtg*: Both agent personality and fatigue has effect on communication initiation.

We perform ten simulation runs for each case and the number of communications measured at five second intervals are averaged. The graph given in Figure 5.4 summarizes the results.

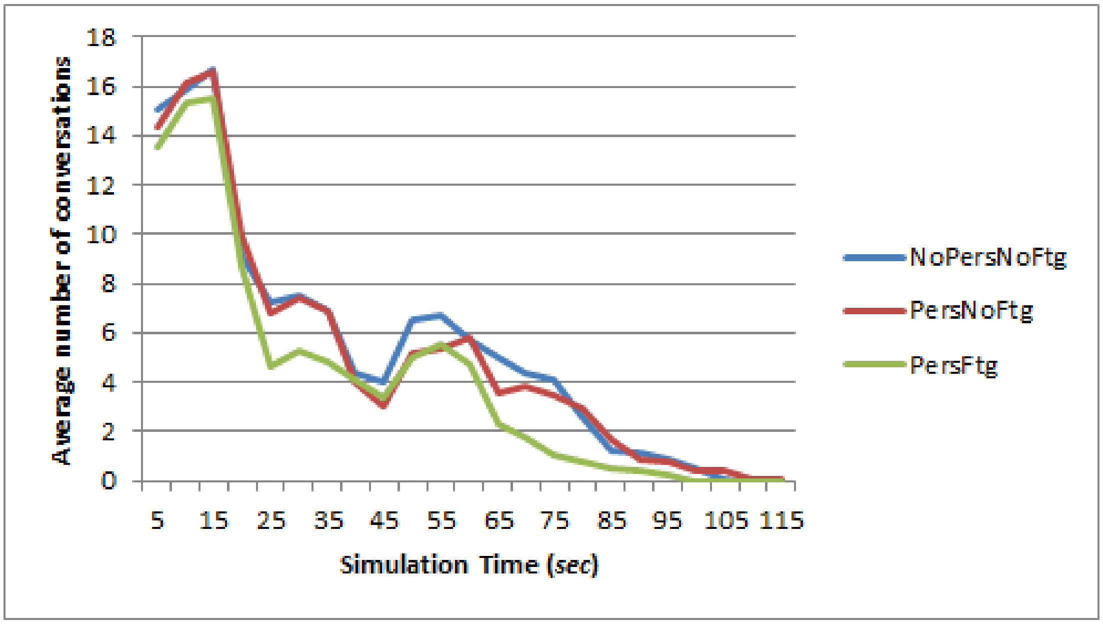


Figure 5.4: The average number of active conversations among ten simulation runs at five second intervals.

5.4 Discussion

It can be observed in the visual simulation results for Experiment Set I Scenario 2 that when the main door becomes unusable, the agents who were targeting this door autonomously begin to form a discussion group. They share information about a new exit route and each individual leaves the group to execute its own plan. When personalities are varied between female and male characters in Experiment Set I Scenario 3, it is observed that the females tend to communicate more, whereas males appear to be eager to leave the discussion group quickly.

The values for the average number of interactions per agent and the standard deviation of these values among agents are given for Experiment Set II in Tables 5.2 and 5.3. The main reason to include varying attributes for agents and simulate their effect on communication is to achieve a level of heterogeneity for communication behavior. In other words, we wanted agent personalities to influence communication, for example, some agents to be more talkative and some to be more reluctant to talk. In our results, although the mean values vary due to the pseudo-random settings and decisions, there is a visible tendency in standard deviation values. For both scenarios, there is an increase in the standard deviation when the personality is varying and effecting communication as compared to the case where the personality is fixed and not effecting communication. This means that when we vary the personalities, the number of interactions vary more among agents. This is a correlation we expected to observe.

In the final set of experiments, we investigate the effect of a single dynamic attribute, fatigue, to communication. When we compare the results for *NoPersNoFtg* and *PersNoFtg* cases with the *PersFtg* case, we observe that *PersFtg* values generally remain below, particularly as simulation time increases. This observation is in line with our expectations because agents become more tired in time and tired agents prefer to communicate less.

Overall, some or perhaps all of these preliminary experiment settings and scenarios may be considered unrealistic. However, this should not be a concern as they are mainly designed to observe and test whether the intended effects are achieved. We have mainly experimented with the attributes' effect on communication initiation. In general, our results seem to approve that static attributes' effect on communication can enhance heterogeneity and dynamic attributes' effect can be used to add dynamic changes in communicative behavior. However, it is important to ask whether the improvement in heterogeneity and the introduction of dynamic changes can enhance the user-perceived realism.

A key realization at this point was that justifying the usefulness of intended communication model and its validation are more crucial issues than constructing a complex model. In other words, there is no point in developing a complex model if its contribution to the perceived realism is to be minor. Therefore, our primary concern has become a search for ways to justify the use of a communication model. In our efforts to do so, two issues turned out to be the major focus points.

1. Various realistic scenarios are required to justify and validate the model as well as to see how it can be improved.

2. To the best of our knowledge, there is no crowd simulation work that focuses

on inter-agent information exchange and its effects on behavior but we still need some form of comparison with existing crowd simulation techniques.

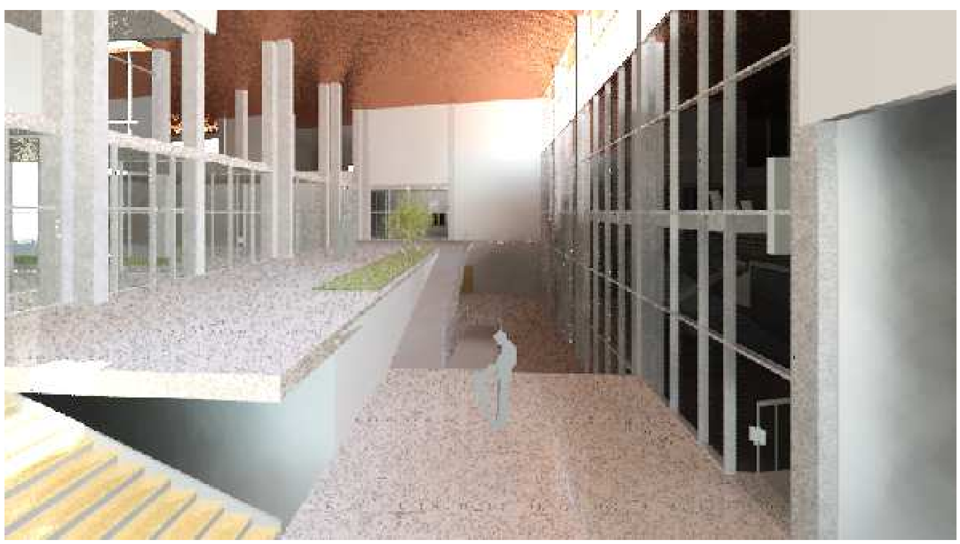
As a result of these issues, we backtracked to look at agent communication standards of MAS community and existing crowd simulators. In particular, we backtracked to FIPA ACL. Earlier, this standard was not considered directly related as it was criticized to be insufficient to simulate human-like interactions. However, we began to be concerned more about evaluation and less about the communication model being human-like. As a result, FIPA ACL specification and KQML that it superseded were explored.

Another problematic issue was the fact that in scenarios we developed, the simulation environments were often unrealistic. Modeling a complex and large environment in detail requires architectural skills and considerable amount of time (especially for the untrained and/or untalented). Acquiring such models is also difficult but thanks to an friend, Architect Mete Sezer, a realistic, large, and complex school building model was obtained. Figure 5.5 shows a simple rendering of this school building model. The actual structure does not exist currently but it is planned to be constructed in the near future.

Prior to using this realistic building, CPGs were used to keep the building connectivity information. It was used in global path planning and most of the communication acts used personal agent CPGs. The CPG for the new building would be large and complex and its creation was found to be very time consuming. Figure 5.6 shows only a small part of the full CPG, which roughly constitutes one fifth of only a single structural level. The automation of the procedure was attempted but the results were unsatisfactory. Due to the lack of CPGs, the global path planning was changed in such a way that it uses a navigation mesh and the contents of inter-agent communication were modified not to include CPGs.

An important question is "How is our work justified and what use will it be?" After the preliminary experiments, there has been a change in how we answer this question. Early on, the intention was to develop a communication model that was going to allow agents to communicate as human-like as possible so that the

(a)



(b)

Figure 5.5: Rendered images of the school building model: (a) a top-down view of the whole model and (b) a view from inside showing the details.

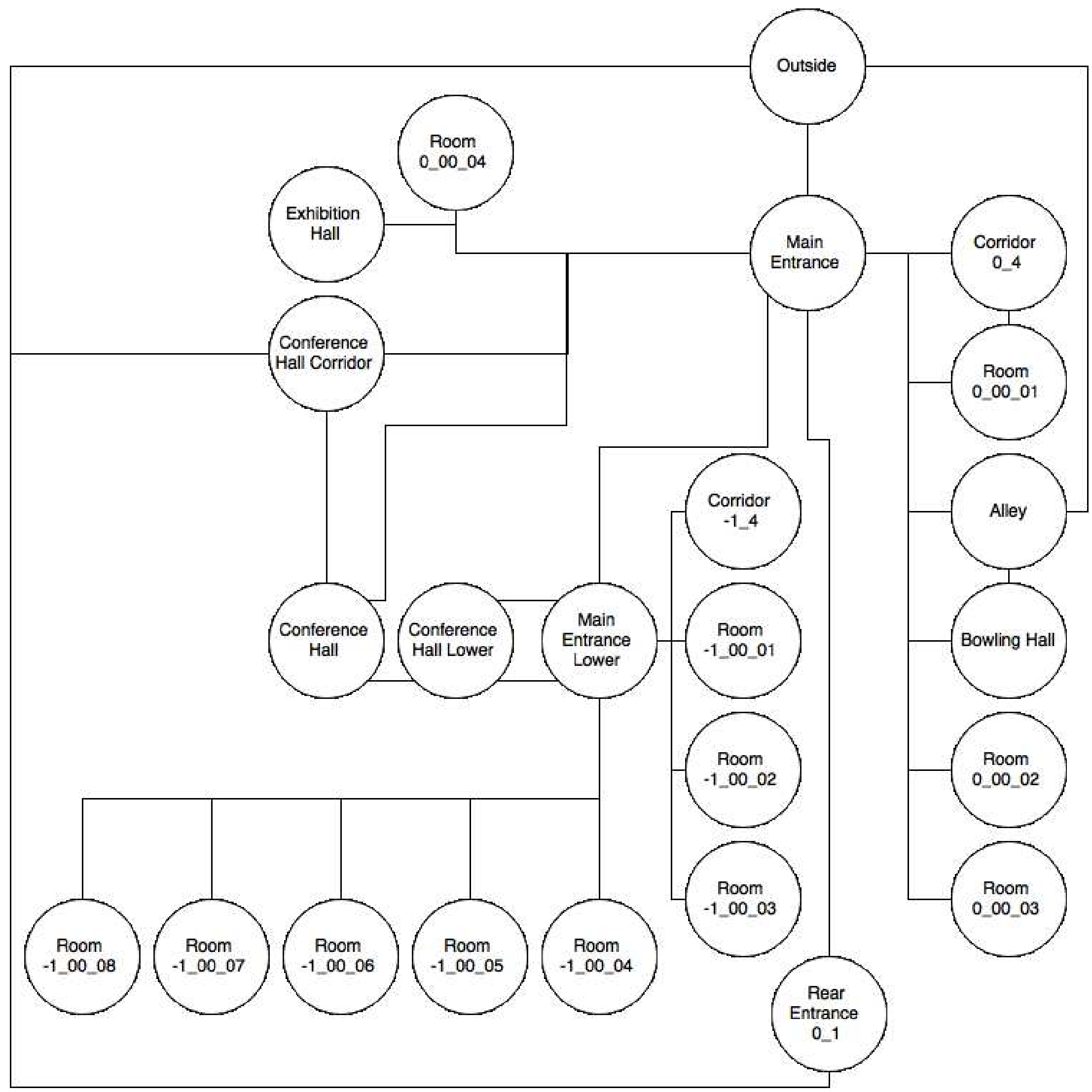


Figure 5.6: A subgraph of the building's Cell-Portal Graph that only shows one adjacency level from the main entrance and the lower part of the main entrance.

realism of the simulated crowd would be improved. Then, it became apparent that considerable improvement in realism will not stem from trying to imitate the complexities of real world human communication. As a result, a divergence from the original intention became necessary. The most important aspects of the target communication model are considered to be the following two notions:

- The model should be complex enough just to facilitate externally observable information passing between agents.
- The model should not be complicated by capabilities which result in very little or no observable difference in simulations.

These new boundaries for the communication model caused some changes in research direction. An example is the way we consider FIPA ACL. We became aware of this standard early in our research. But, it was not considered because it was criticized for being inadequate to simulate human-like interactions between agents. When the intention for the communication model became not to be as human-like as possible, FIPA ACL was reevaluated for usefulness. Making the agents FIPA ACL compliant or adopting a subset of FIPA ACL to regulate messaging mechanisms was found to provide some advantages.

A second question that is persistently difficult to answer in crowd simulation research is "How can we validate the results?" The general aim of the field is to simulate visually and/or behaviorally ever realistic crowds of as many agents as possible as fast as possible. The number of agents and the simulation frame rates are quantitative properties, and therefore, easy to compare. The visual realism is difficult to quantize as well but the hardest property to validate is the behavioral realism. Efforts on automated analysis of real-world crowd videos and comparison attempts of virtual with real crowd behavior exist but a generally applicable validation method currently does not.

A more common method in the field is to use comparisons between simulations. Generally, simulations generated using the novel approach in question are compared with others generated using alternative approaches. In the case that an alternative does not exist, the comparison is carried out with simulations generated by disabling the novel approach. In fact, one of the motivations for developing crowd simulation frameworks such as ADAPT and Menge reviewed in Section 2.1 is to facilitate meaningful and better controlled comparisons between different approaches.

Various scenarios can be authored to show the effect of the communication model but how will we evaluate the realism of resulting simulations? The options are (i) to compare quantitative simulation results with real world data, (ii) to analyze quantitative simulation results and look for conformances with existing theories, and (iii) to analyze visual simulation results and look for similarities with real crowds. The first two options are more objective and scientifically correct whereas the last one is more subjective.

There exist several works aimed at extracting data from real crowd videos such as [66], [67], [68], and [69]. The extracted data are often the movement trajectories of the individuals in the video. The problem (in terms of the first evaluation option) is that it is not clear how the effect of communication can be understood from trajectories.

Realizing the second evaluation option has proven even more difficult. We have browsed psychology and sociology literature to a certain degree but were unable to locate a general work on effects of intra-group communication on group behavior. Nonetheless, the search has not been completely useless. It was learned that people generally panicking in crowd situations is a popular but false assumption [70]. On the contrary, individuals experiencing a crowd incident mostly behave rationally but inadequate information is often a factor in most incidents.

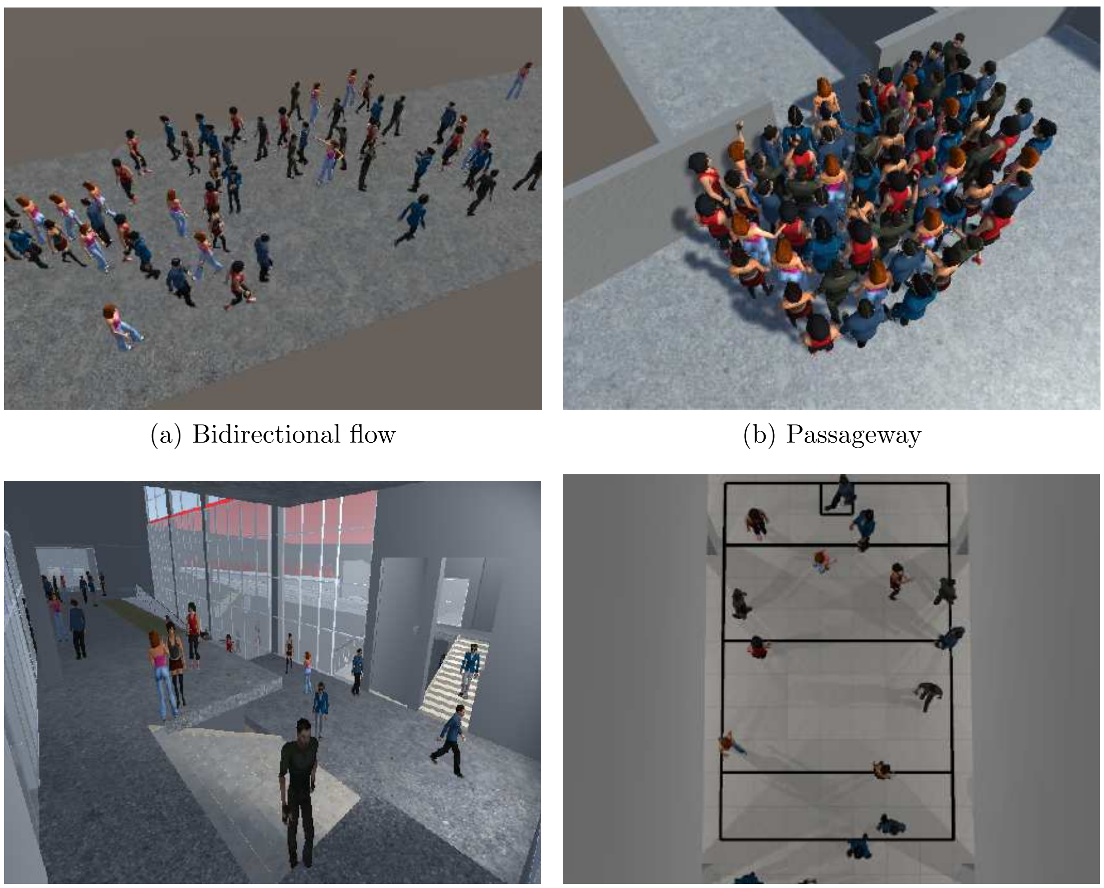
The third option for evaluation appears to be the most applicable one. Although subjective in nature, it is still useful to show the effect of an applied technique or a new model. In particular, when the crowd simulation's main focus is not navigation but to incorporate a human factor, this third option becomes the major evaluation method.

Chapter 6

Simulation Scenarios and Evaluation

The final version of our communication model is shown in action by four example scenarios (cf. Figure 6.1) described below. There is only hollow communication in the first two scenarios. The evacuation of a realistic building model forms the third scenario. Here, the simulated communications are meaningful instances in the form of asking and answering about exit routes. In the last scenario, we compare simulated agent trajectories with real pedestrian trajectories extracted from a video. We used the Unity® Game Engine [71] to realize all these simulation scenarios.

It was previously mentioned that the lower parts of the navigation component in Figure 3.1 are scenario independent. This means that it does not make a difference whether the destination given by the high-level planning is an intermediate target or a final one. An agent moves from her/his current position to any location by querying the static and global navigation mesh and then locally avoiding collisions while following the calculated path. We make use of the built-in game engine tools to achieve these. The preferred speed is the only agent attribute related to these tasks.



(a) Bidirectional flow (b) Passageway (c) Evacuation (d) Chat

Figure 6.1: Simulation screenshots from the four example scenarios used. There is only hollow communications (i.e., communications without important information sharing,) in the bidirectional flow, passageway, and chat scenarios. Meaningful communications by which agents share environment and exit route take place in the evacuation scenario.

The communication component of our agent architecture also has both scenario-dependent and scenario-independent parts as described in the previous chapter. General and low-level tasks constitute the AV layer whereas scenario-dependent tasks are in the FoE layer. Essentially, when one wants to use our communication model in a new scenario, the primary task is to define the necessary parts in the FoE layer. In particular, the new message types and the corresponding high-level behavior for each such type should be determined.

6.1 Bidirectional Flow and Passageway Scenarios

These two relatively simpler scenarios required a straightforward high-level planning in navigation. It merely constitutes of a single destination computation. In the bidirectional flow scenario (cf. Figure 6.1a), an agent at one end of the street calculates a destination position at the other end. All agents in the passageway scenario (cf. Figure 6.1b) try to reach a location at the other side of the passageway a little further down the short corridor.

Both of these scenarios include only hollow communication. Therefore, CHAT is the only FoE layer message type. In order to apply the communication model in these scenarios, we determined the related high-level behavior. This involves when and how a CHAT message will be sent by an agent and what a receiver will do upon receiving one. These mechanisms involve two parameters, namely *communication probability* (p_c) and *walk and talk ratio* (p_r). These are used to control when and how a CHAT message is sent. When the following four conditions are satisfied, an agent starts a communication with probability p_c.

i) This agent (the sender) is not already communicating;

ii) There is some other agent in the hearing range (called the receiver below);

iii) This agent did not communicate with the receiver before;

iv) The receiver is not currently communicating.

When these four conditions are satisfied, the agent sends a CHAT message with probability p_c. This is achieved by pseudo-randomly generating a value in the $[0, 1]$ range and using a less-than-or-equal-to comparison of this pseudo-random value with p_c. In the case that the comparison is true, the second parameter p_r which is also in the range $[0, 1]$ becomes important. p_r is the tendency for an agent to talk while walking. When 0, agent always prefers to stop and talk and when 1, agent always prefers walking and talking. Intermediate values represent intermediate likelihoods. A second pseudo-random value is used in the same manner but this time using p_r. When walking and talking is preferred, the receiver needs to be walking in approximately the same direction. If all these conditions are satisfied, the agent forms a CHAT message, adjusts its speed according to the receiver's, and AV layer takes control to send the message. On the other hand, agents communicate by first stopping and turning towards each other if the comparison with p_r returns false.

We collected flux data in the passageway scenario so that a comparison can be made with the data in [72]. This dissertation proposes a psychological and sociological perspective of human behavior in emergencies and includes development of a system called MASSEgress. Pan repeats the passageway scenario of a previous work using Simulex [73] for validation. In this scenario, 100 agents are positioned in a $5m \times 5m$ area and there is a single passageway to exit. By varying the passageway, the flux rates are recorded. The results are compared with those acquired earlier using Simulex. In our simulation, the flux rates are recorded in the same manner for a similar comparison. At each width, the simulation is carried out twice: once with no communication, and then, with communication. Figure 6.2 presents the results.

As the passageway width increases and exiting becomes easier, the flux values converge towards values that depend on the preferred speeds instead of increasing. This is because they are computed with the number of exiting people per second and per unit width. All simulation results display the same convergence behavior towards the right side of the chart in Figure 6.2. The results differ more on the left

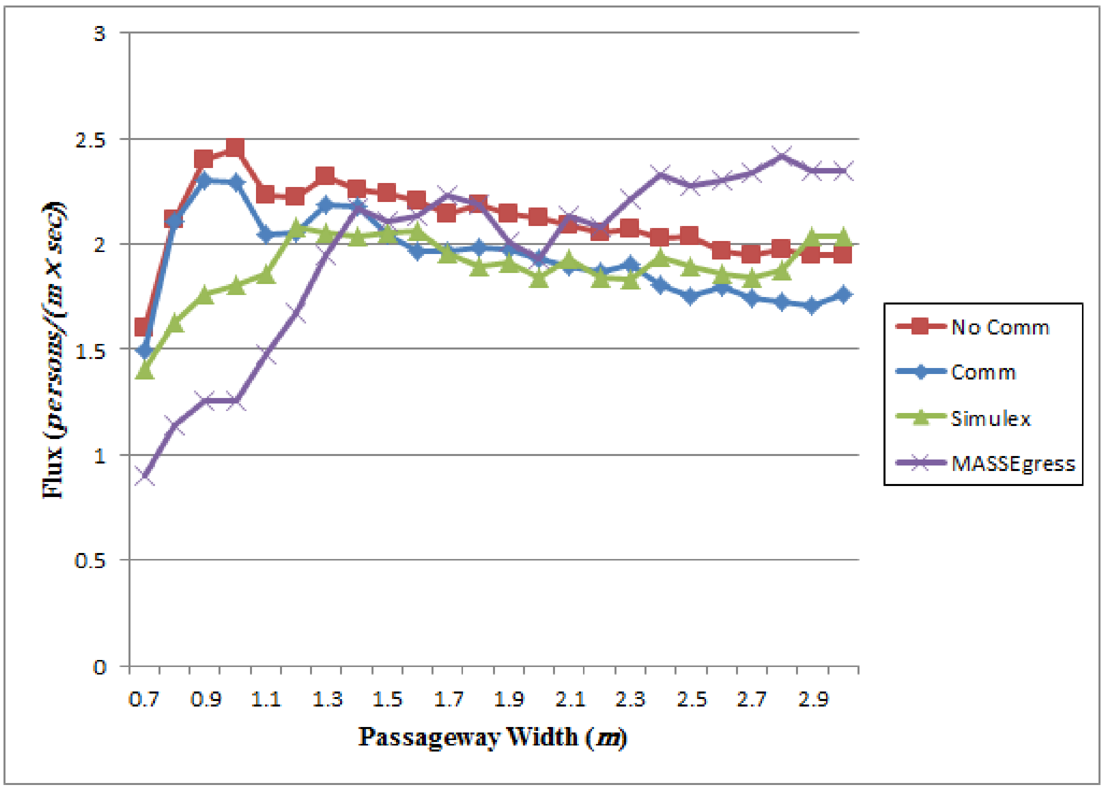


Figure 6.2: The passageway scenario flux comparison. It should be noted that the values are flux values (persons/($m \times$ sec)), not flow rates (persons/sec). The flux does not increase as the passageway width increases because it is calculated as 'per unit width'. Instead, it converges to a preferred speed-dependent value. Our results are labeled as *No Comm* and *Comm*. The results from [72] are labeled as *Simulex* and *MASSEggress*.

side of the chart when the passageway width is small. Congestion becomes worse with a narrow passageway and this amplifies the effect of using different local collision avoidance methods. Nonetheless, the results are overall not very different from the results obtained in MASSEgress and Simulex. It is also important to note that communication did not influence and change the flux marginally.

6.2 Evacuation Scenario

There are important differences between the evacuation scenario (Figure 6.1c) and the other scenarios. One of these differences is in the high-level planning logic of the agents' navigation components. This part contains a secondary contribution of this dissertation, which is an algorithm to model the evacuation behavior for an agent that does not know the environment. Figure 6.3 describes the planning algorithm. Whether an agent knows the environment is represented with a boolean attribute. An agent that has prior environment knowledge can directly choose the best emergency gathering area. Notion of one area being better than another can be described by having a shorter path to it. When this is the case for an agent, (s)he will not ask others about how to exit but others can initiate a communication with her/him. In the opposite case, i.e., when the agent has no prior information about the environment, the first option is to go for a seen exit if there is one. We found direction sign following to be the next appropriate agent behavior. It should be noted that the implementation allows us to enable the effect of direction signs when starting a simulation run. Communication with other agents becomes the next best option when either direction signs are disabled or no signs are in sight. Yet, unless communication is possible, navigation continues to be in control. The planning algorithm also considers following behavior. This becomes a frequently used method when communication capability is turned off. An agent initiates exploration when none of the aforementioned actions are possible.

As long as the agent is in NOT_COMMUNICATING or WANTS_TO_COMMUNICATE states, the planning algorithm in Figure 6.3 is in control. But the navigation

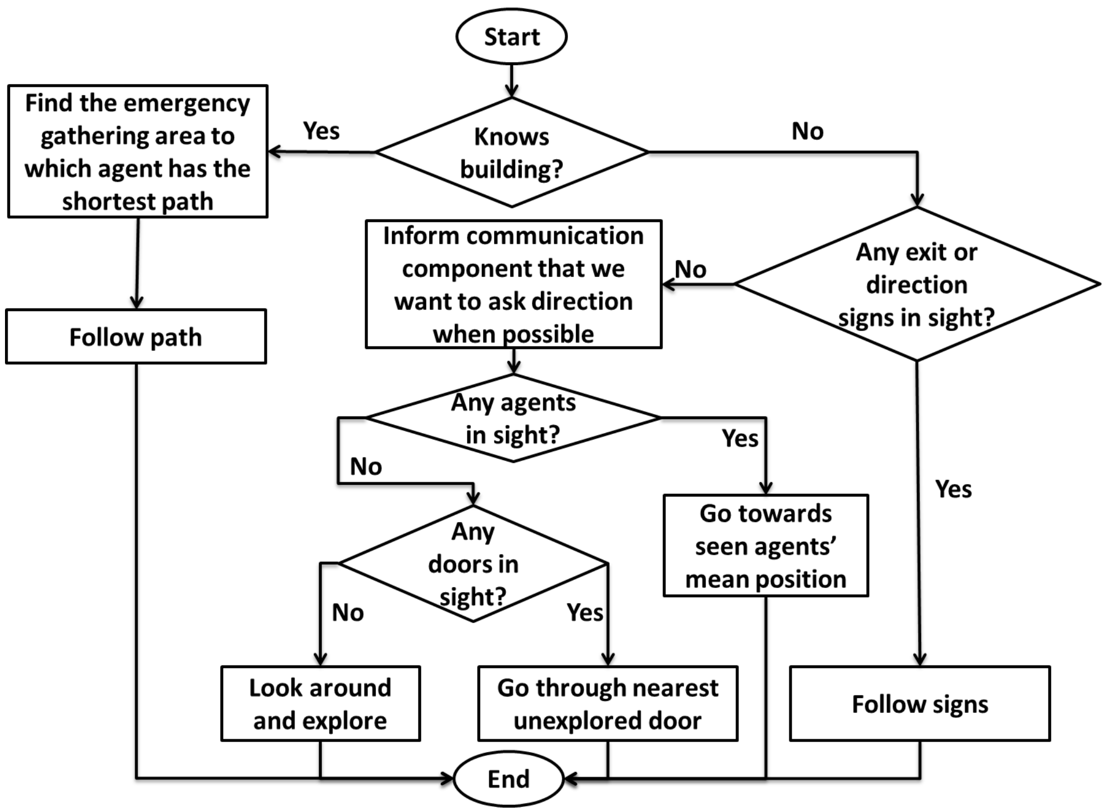


Figure 6.3: The high-level planning algorithm developed for the evacuation scenario. A boolean agent attribute represents whether (s)he knows the environment. Use of direction signs can be enabled/disabled. Communication is not a requirement for this planning algorithm. Agents without a priori environment knowledge can display plausible evacuation behavior based solely on this algorithm. But, combining this with the communication component allows us to simulate meaningful communication.

is temporarily stopped and the communication component takes control when the agent finds a target for communication or when a message is received. The communication instance will finalize possibly causing a change in agents' knowledge and the navigation component will regain control.

We initiate a simulation run by positioning a given number of agents randomly on the building's navigable areas. There are six different simulation settings we used.

- *Only Nav (Know=0):* Both agent communication and use of direction signs are turned off. Agents' navigation capability is the only mechanism for them to evacuate. Agents do not have apriori environment knowledge. This setting can be considered as the worst case in terms of efficient evacuation.
- *Only Nav (Know=0.5):* Differs from the previous setting at only one point. This time, each agent has 0.5 probability to have environment knowledge when being initiated (compared to 0 probability in the previous setting). The net effect is that stochastically 50% of the agents know the building.
- *Comm:* Again there is a single difference with the previous setting. Agent communication is enabled. Direction signs are still turned off and environment knowledge probability is 0.5.
- *Sign:* This time, the effect of direction signs are enabled but agent communication is turned off. The probability for an agent to have prior environment knowledge is still 0.5.
- *Comm&Sign:* In this setting, we turn on both direction signs and agent communication. The environment knowledge probability is same as the previous three settings, which is 0.5.
- *Only Nav (Know=1):* This setting is similar to the first two settings in the sense that both direction signs and agent communication are turned off. The difference of this setting is that the probability for an agent to have environment knowledge is 1. This means that all agents know the building. So, this setting in a way represents the best case.

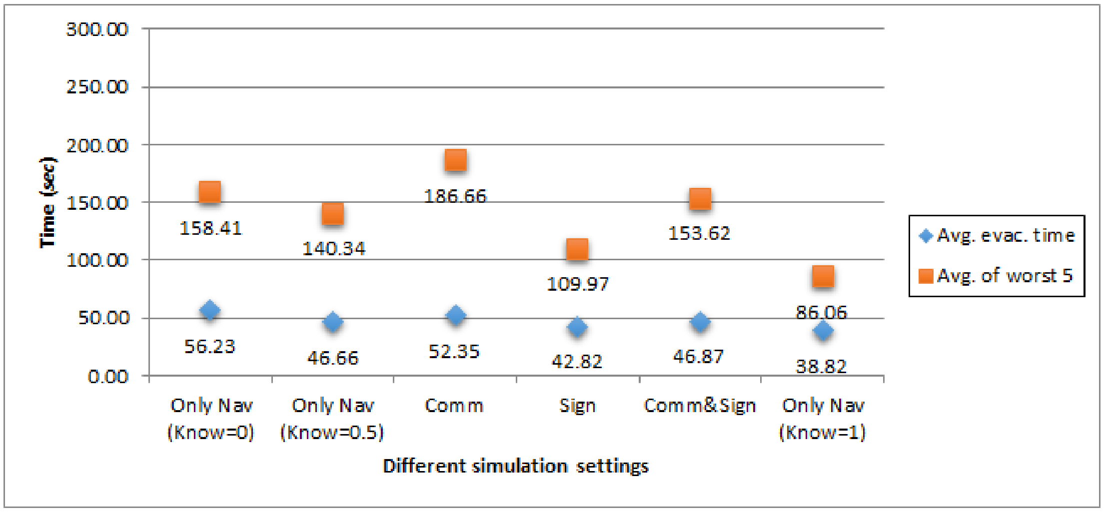


Figure 6.4: Average evacuation times measured in the evacuation scenario for 50 randomly placed agents. There are six different simulation settings: Only Nav (Know=0), Only Nav (Know=0.5), Comm, Sign, Comm&Sign, and Only Nav (Know=1). The simulation is carried out five times for each setting. *Avg. evac. time* (blue) is the average of all agents' evacuation times over all five executions. *Avg. of worst 5* (orange) is the average of the five largest (i.e., worst) evacuation times. The setting names are explained in the text in detail.

Our measurements for agent evacuation times are summarized in Figures 6.4, 6.5, and 6.6. We used the blue marks to show the evacuation time averaged over all agents after carrying out the simulation five times using each setting. The orange markers show the average of the five largest (i.e., worst) evacuation times. What we meant by worst and best case for the simulation settings is clear from the leftmost and rightmost values in all graphs. Remember that in the *Only Nav (Know=0)* setting, agent communication and direction signs are disabled and none of the agents know the building. As a result, their evacuation planning is based on seeing an exit, following others, and exploration. In the first columns of all three charts, the average evacuation times (blue) are high as expected meaning that it takes agents longer to evacuate. The same is also valid for orange values (the averages of the five largest evacuation times). On the opposing end, the *Only Nav (Know=1)* setting means that everyone knows the ideal path to exit the building. Hence, it is natural to see in the last columns that both averages are relatively small.

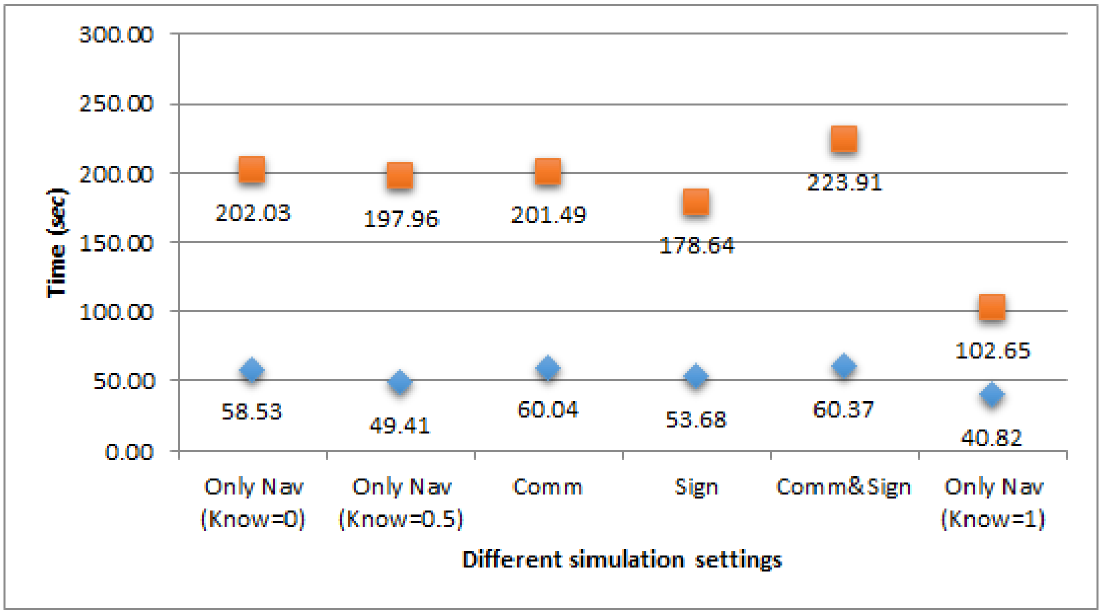


Figure 6.5: Average evacuation times measured in the evacuation scenario for 100 randomly placed agents. Same as Figure 6.4, except the number of agents.

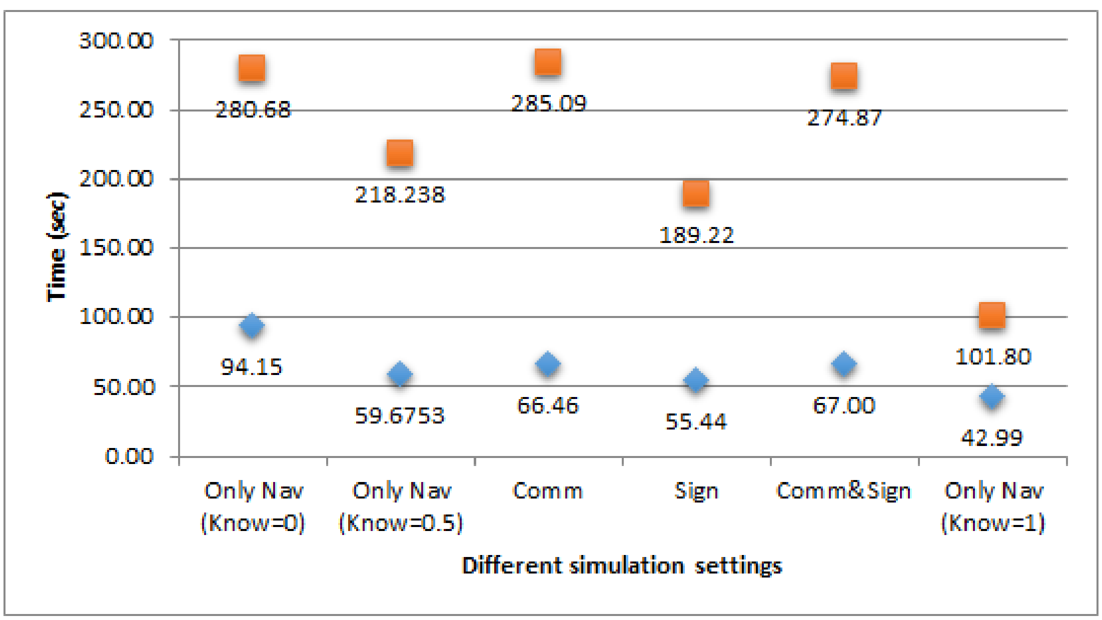


Figure 6.6: Average evacuation times measured in the evacuation scenario for 200 randomly placed agents. Same as Figures 6.4 and 6.5, except the number of agents.

In all other intermediate settings, stochastically 50% of agents have prior environment knowledge. As a result, the values are in between those on each side. Our expectations are apparent in the difference of values between different settings. To begin with, it is observed that in general, the evacuation times increase as the agent count is increased. This is expected as more agents cause more congestion. Secondly, *Sign* setting appears to produce relatively smaller evacuation times. This is also expected since direction signs help agents to evacuate. Thirdly, the *Comm* setting results, especially the orange values, seem to be relatively larger which means that evacuation times are increased due to agent communication. This is also natural as agent communication requires agents to occasionally stop and talk.

We can test whether stopping and talking is truly the reason for this increase in evacuation times by looking at the travel distances. If travel distances in the *Comm* and *Comm&Sign* settings are not significantly greater than those in the *Only Nav (0.5)* and *Sign* settings, then we can say that the increase is due to stopping and talking. The ten longest trajectories in each setting were measured (cf. Figure 6.7). The values with *Comm* and *Comm&Sign* settings turned out to be not larger but smaller compared to those with (*Only Nav (0.5)* and *Sign*) settings. These average lengths together with the previous average evacuation times proves that when agent communication is turned on, agents travel less but their evacuation times increase.

Some running time measurements for the evacuation scenario with varying number of agents are given in Table 6.1. These measurements were recorded while simulations were carried out on a PC with an Intel Xeon E5-2643 3.3GHz processor, 48GB RAM, and a Nvidia Quadro 4000 graphics processor. The smallest and largest values for the time in between two consecutive frames are given as minimum and maximum frame times. Additionally, the frame rate (the number of frames per second; fps) is averaged over the simulation run and these values are presented in the last column. It was realized that although the exact contribution varied depending on what falls into camera view, rendering (drawing) was often responsible for most of the computation. To account for this fact, performance measurements are carried out when there are zero agents in the simulation and

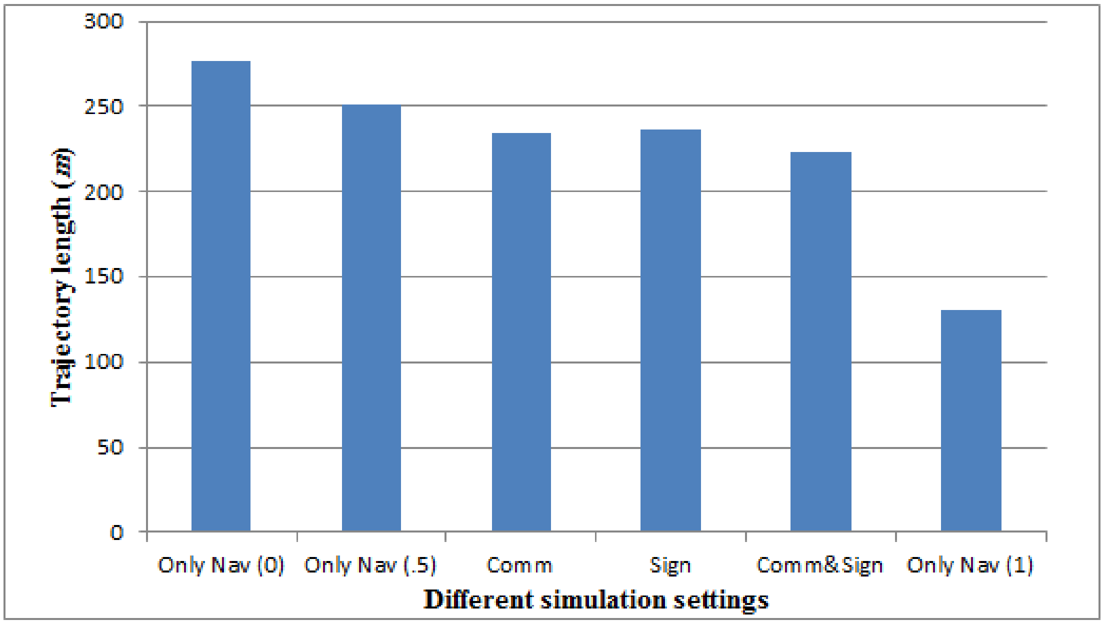


Figure 6.7: The average lengths of ten longest trajectories in each simulation setting.

these measurements are included in the table. Zero agents mean no perception, navigation, or communication calculations. The average frame rate was 27.42 fps even when there were no agents. The values measured when there are agents should be evaluated with respect to these baseline values for zero agents.

6.3 Chat Scenario

We use this final scenario in order to compare the trajectories and behaviors generated by our algorithm with those from a real-world scenario. A video that involves pedestrians occasionally chatting (cf. Figure 6.8a) is chosen from a real crowd video data collected by the Movement Research Lab at the Seoul National University [66]. We extracted the movement trajectories of the pedestrians from the video. We also constructed a 3D model of the environment observed in the video in Unity (see Figure 6.8b). We place agents at positions corresponding to the initial positions in the real video. We carry out simulations both with and without agent communication. Agents continuously choose random targets

#agents	Min frame time (sec)	Max frame time (sec)*	Avg. frame rate (fps)
0	.016558	.325687	27.42
50	.016566	.333333	25.12
100	.016559	.333333	21.61
200	.026999	.333333	13.53

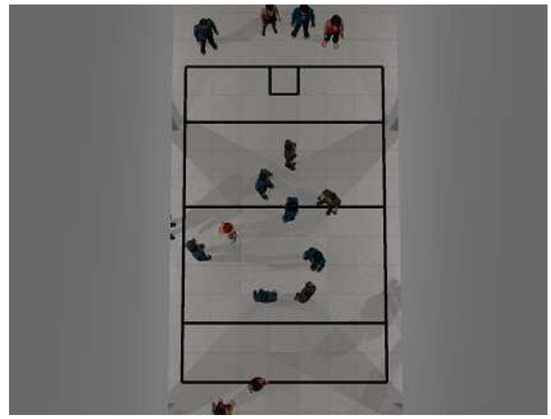
Table 6.1: Some performance measurements with different number of agents in the evacuation scenario. The values indicate the computational cost of agent architecture (navigation, perception, and communication together) when considered relatively against those in the first row. The first row with no agents should be taken as a baseline that gives a clue about the rendering cost of the environment. The results are recorded using the *Comm* setting. (*) There is a limit on maximum frame time in Unity game engine to prevent freezing. 0.333333 values correspond to this limit, which ensures three fps.

and navigate to these locations. We extracted the simulated agent movement trajectories by recording positions at one second time intervals.

In order to compare the real and simulated trajectories, we used vfractal estimation [12]. The range $[0.1, 10]$ is divided into 100 different step sizes with 0.1 increments. The unit of length here is the size of a video pixel. We scaled all simulation coordinates (and therefore lengths) to match this unit. Floor geometry enabled us to compute the scaling parameters for both coordinate dimensions.

Same vfractal calculations are carried out for (i) the real video trajectories (blue), (ii) the simulated trajectories with no agent communication (red), and (iii) the simulated trajectories with agent communication (green). Figure 6.9 summarizes the results of the estimations. It is observed in the results that for both the estimations (solid lines) and the confidence in these estimations (dashed lines), the green values are closer to the blue values than the red values are to the blue values. This means that the straightness/crookedness of the simulated agent movement trajectories better match that of real trajectories when agent communication is turned on. Lastly, the averages of vfractal values and confidences over different step sizes are computed to summarize our results (see Table 6.2).

(a) Real video



(b) Simulation

Figure 6.8: Chat scenario: a still frame from (a) the real video and (b) the virtual simulation scene.

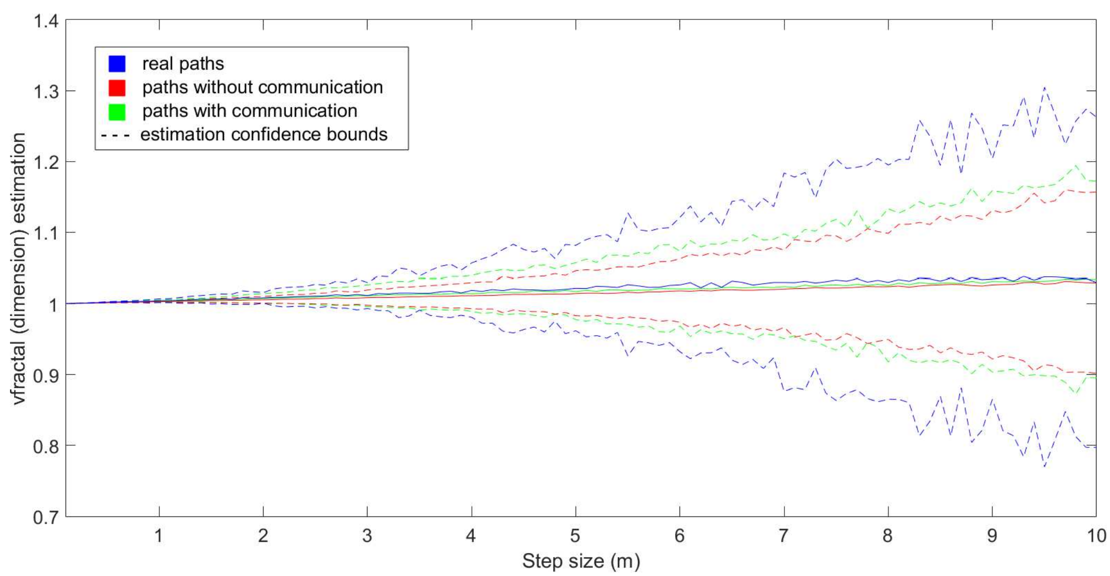


Figure 6.9: Vfractal estimations and confidence bounds for real video pedestrian trajectories (blue), simulated trajectories when agents do not communicate (red), and simulated trajectories when agents can communicate (green). The estimated fractal dimension (d) is the vertical axis and the step size (scale) used in the estimation is the horizontal axis. The actual estimations are represented with the solid lines and the confidence bounds in these estimations are shown with the dashed lines.

Trajectory	Average vfractal	Confidence
Real	1.0209	± 0.0871
Simulated (Comm)	1.0178	± 0.0521
Simulated (NoComm)	1.0150	± 0.0421

Table 6.2: Mean vfractal estimations and confidences calculated by averaging the values in Figure 6.9 over the range of different step sizes.

6.4 Analysis

We can observe plausible autonomous behaviors in simulations. Clearly, when agents not only walk around but also autonomously communicate with each other, the behavioral variety is enhanced. Our model only considers communication between two agents at a time but multiple such communications occasionally happen at the same time and it looks like discussion groups are formed autonomously. This is in line with with the recent understanding that people in a crowd mostly move as a group rather than as individuals [74]. The evacuation simulation was important for simulating meaningful communications. The measurements carried out in this scenario showed us that the influence of communication on behavior is consistent with the expectations. In the last scenario, the vfractal results showed that the straightness/crookedness of the simulated trajectories fit the real trajectories better when agents communicate.

Other than the perception component, our additions to agent architecture to enable communication are simple enough and they contribute little to the computational cost. The current implementation for the perception subcomponents function like collision detection methods. These affect the computation time considerably. However, it should be possible to improve and optimize the perception implementation Also, with the application of space partitioning or level-of-detail techniques, it should be possible to achieve interactive rates for higher number of agents.

Chapter 7

Conclusion

A primary aim in the field of virtual crowd simulation is to obtain plausible autonomous behavior. One cannot deny that communication takes place in the real-world crowds. As a result, we set out to model deliberate inter-agent information exchange in virtual crowds and investigate its effects on virtual crowd behavior. We stayed away from having to deal with subtleties of human behaviors and languages. To achieve this, we took communication and information concepts at an abstract level, not in specific forms such as speech or sentences.

A message structure based on the FIPA ACL message structure is used in our communication model. Realizing communication behavior is achieved with a PDA implementation. By combining the communication model with a perception and navigation implementation, the development was employed in four example scenarios. The dissertation shows example visual simulation outputs as well as providing an overview and emphasizing important issues.

An important aim for us was to come up with a design which is easy to apply in new scenarios. In order to achieve this, a layered design where the low-level, message type and scenario-independent tasks (AV layer) are separated from those that are high-level and scenario-dependent (FoE Layer), is preferred. When a user wants to apply this model in a new scenario, the only requirement is to define the necessary scenario-specific message types and high-level behavior related to sending and receiving these types of messages.

We not only showed visual simulation outputs, but also provided various quantitative measurements (flux, evacuation times, trajectory lengths, and trajectory shapes) and base our evaluation on these measurements. The improvement in behavioral variety and example plausible behaviors can be observed by watching the visual outputs. We draw two conclusions from the flux comparison in the passageway scenario. First, it is shown that the navigation mechanisms we used behave similarly to the existing systems. Second, communication does not disturb the passageway flow in any significant way.

An evacuation scenario was used in order to simulate meaningful communication, i.e., communication that influences behavior. A separate algorithm was needed to model the evacuation behavior for an agent that does not know the environment. Our expectations were not contested by the average evacuation time measurements. When agents use direction signs, they evacuate faster however, when they communicate with each other, they slow down. On the other hand, the lengths of the longest trajectories, showed a decrease in travel distance with agent communication. The logical combination of these two results is that when asking and answering about direction is simulated, agents traveled less but took more time to evacuate.

We carried out vfractal estimations in the final chat scenario so that simulated trajectories can be compared with real trajectories extracted from a video. We saw in this set of results that simulation trajectories are closer to real trajectories in terms of straightness/crookedness when agent communication takes place.

Through some time measurements, we showed that the additional cost of simulating agent communication is not significant with respect to the the overall cost of multi-agent simulation. Moreover, it was realized that the perception model we used was responsible for most of this overhead. Improvements to the perception implementation are possible and the good news is that changes made to it should not affect the communication model much as a result of modular design.

In addition, space partitioning and/or level-of-detail techniques could improve the performance even further.

There are some issues that we could not investigate but remain as possible extensions to this dissertation. One issue is that user studies can be used to compare the plausibility of the video outputs with and without communication. A second method for better evaluation can be to develop similarity metrics for comparing crowd behavior and making use of these metrics to compare simulation results and real crowd videos. Other possible extensions are improving the efficiency of the perception component, introducing a new intermediate communication layer responsible for dialog management, combining the communication model with more sophisticated agent architecture, and applying the model in other scenarios. We plan to investigate some of these possible extensions.

Bibliography

[1] P. Harding, S. Gwynne, and M. Amos, "Mutual information for the detection of crush," *PLoS One*, vol. 6, no. 12, pp. 1–10, 2011.

[2] A. Bera, T. Randhavane, R. Prinja, and D. Manocha, "Sociosense: Robot navigation amongst pedestrians with social and psychological constraints," *arXiv preprint arXiv:1706.01102*, 2017.

[3] R. T. Hays and M. J. Singer, *Simulation Fidelity in Training System Design: Bridging the Gap Between Reality and Training*. New York, NY: Springer-Verlag, 2012.

[4] S. J. Guy, S. Kim, M. C. Lin, and D. Manocha, "Simulating heterogeneous crowd behaviors using personality trait theory," in *Proceedings of the ACM Symposium on Computer Animation (SCA)*, (Vancouver, Canada), pp. 43–52, ACM, 2011.

[5] N. Pelechano, J. M. Allbeck, and N. I. Badler, *Virtual Crowds: Methods, Simulation, and Control*. Synthesis Lectures on Computer Graphics and Animation #8, San Rafael, CA: Morgan & Claypool Publishers, 2008.

[6] F. Durupınar, U. Güdükbay, A. Aman, and N. I. Badler, "Psychological parameters for crowd simulation: From audiences to mobs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 9, pp. 2145–2159, 2016.

[7] Q. Yu and D. Terzopoulos, "A decision network framework for the behavioral animation of virtual humans," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, (San

Diego, CA), pp. 119–128, ACM SIGGRAPH/Eurographics Association, 2007.

[8] J. Funge, X. Tu, and D. Terzopoulos, “Cognitive modeling: knowledge, reasoning and planning for intelligent characters,” in *Proceedings of the ACM SIGGRAPH International Conference on Computer Graphics and Interactive Techniques*, (Los Angeles, CA), pp. 29–38, 1999.

[9] S. Kim, S. J. Guy, D. Manocha, and M. C. Lin, “Interactive simulation of dynamic crowd behaviors using general adaptation syndrome theory,” in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (i3D)*, (Costa Mesa, CA), pp. 55–62, ACM, 2012.

[10] P. Watzlawick, J. B. Bavelas, D. D. Jackson, and B. O’Hanlon, *Pragmatics of Human Communication: A Study of Interactional Patterns, Pathologies and Paradoxes*. New York, NY: W. W. Norton, 2011.

[11] S. Poslad, “Specifying protocols for multi-agent systems interaction,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 2, no. 4, 2007.

[12] V. O. Nams, “The vfractal: a new estimator for fractal dimension of animal movement paths,” *Landscape Ecology*, vol. 11, no. 5, pp. 289–297, 1996.

[13] S. Ali, K. Nishino, D. Manocha, and M. Shah, eds., *Modeling, Simulation and Visual Analysis of Crowds*, vol. 11 of *The International Series in Video Computing*. New York, NY: Springer-Verlag, 2013.

[14] D. Thalmann and S. R. Musse, *Crowd Simulation*. London, UK: Springer-Verlag, 2 ed., 2013.

[15] L. Henderson, “On the fluid mechanics of human crowd motion,” *Transportation Research*, vol. 8, no. 6, pp. 509–515, 1974.

[16] D. Helbing and P. Molnar, “Social force model for pedestrian dynamics,” *Physical Review E*, vol. 51, no. 5, p. 4282, 1995.

[17] V. Blue and J. Adler, “Cellular automata microsimulation of bidirectional pedestrian flows,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1678, pp. 135–141, 1999.

[18] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Pasadena, CA), pp. 1928–1935, 2008.

[19] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha, "PLEdestrians: a least-effort approach to crowd simulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, (Madrid, Spain), pp. 119–128, ACM SIGGRAPH/Eurographics Association, 2010.

[20] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[21] D. Thalmann, "Populating virtual environments with crowds," in *Proceedings of the ACM International Conference on Virtual Reality Continuum and Its Applications (VRCIA)*, (Hong Kong), p. 11, ACM, 2006.

[22] R. McDonnell, M. Larkin, S. Dobbyn, S. Collins, and C. O'Sullivan, "Clone attack! perception of crowd variety," *ACM Transactions on Graphics*, vol. 27, pp. 26:1–26:8, Aug. 2008.

[23] R. McDonnell, F. Newell, and C. O'Sullivan, "Smooth movers: perceptually guided human motion simulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, (San Diego, CA), pp. 259–269, ACM SIGGRAPH/Eurographics Association, 2007.

[24] F. Durupınar, N. Pelechano, J. M. Allbeck, U. Güdükbay, and N. I. Badler, "How the Ocean personality model affects the perception of crowds," *IEEE Computer Graphics and Applications*, vol. 31, pp. 22–31, May/June 2011.

[25] N. Pelechano, *Modeling realistic high density autonomous agent crowd movement: social forces, communication, roles and psychological influences*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 2006.

[26] N. Pelechano, J. M. Allbeck, and N. I. Badler, "Controlling individual agents in high-density crowd simulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, (San Diego, CA), pp. 99–108, ACM SIGGRAPH/Eurographics Association, 2007.

[27] M. Jaros, M. Di Angelo, and P. Ferschin, "Modeling and simulation of pedestrian behaviour: As planning support for building design," in *Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*, (Lisbon, Portugal), pp. 1–8, 2016.

[28] C. Turkay, E. Koc, and S. Balcisoy, "Integrating information theory in agent-based crowd simulation behavior models," *The Computer Journal*, vol. 54, no. 11, pp. 1810–1820, 2011.

[29] B. G. Silverman, M. Johns, J. Cornwell, and K. O'Brien, "Human behavior models for agents in simulators and games: Part I: Enabling science with PMFserv," *Presence: Teleoperators & Virtual Environments*, vol. 15, pp. 139–162, April 2006.

[30] B. G. Silverman, G. Bharathy, and K. O. J. Cornwell, "Human behavior models for agents in simulators and games: Part II: Gamebot engineering with PMFserv," *Presence: Teleoperators & Virtual Environments*, vol. 15, pp. 163–185, April 2006.

[31] R. Narain, A. Golas, S. Curtis, and M. C. Lin, "Aggregate dynamics for dense crowd simulation," *ACM Transactions on Graphics*, vol. 28, no. 5, p. 8 pages, 2009.

[32] F. Qiu and X. Hu, "Modeling group structures in pedestrian crowd simulation," *Simulation Modelling Practice and Theory*, vol. 18, no. 2, pp. 190–205, 2010.

[33] A. Shoulson, N. Marshak, M. Kapadia, and N. I. Badler, "ADAPT: the agent development and prototyping testbed," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (i3D)*, (Orlando, FL), pp. 9–18, ACM, 2013.

[34] S. Curtis, A. Best, and D. Manocha, "Menge: a modular framework for simulating crowd movement," tech. rep., Department of Computer Science, The University of North Carolina-Chapel Hill, 2014.

[35] A. Shoulson, F. M. Garcia, M. Jones, R. Mead, and N. I. Badler, "Parameterizing behavior trees," in *Motion in Games* (J. Allbeck and P. Faloutsos, eds.), pp. 144–155, Heidelberg, Germany: Springer-Verlag, 2011.

[36] E. Lobo-Hernández, X. Luo, G. Alomía-Peñafiel, N. Liu, and C. Zúñiga Cañón, "How parallelization helps crowd simulation: Study of an OpenMP-based system," in *Proceedings of the International Conference on Virtual Reality and Visualization (ICVRV)*, (Zhejiang, China), pp. 354–357, 2016.

[37] R. Craig, "Communication theory as a field," *Communication Theory*, vol. 9, pp. 119–161, May 1999.

[38] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Champaign, IL: University of Illinois Press, 1949.

[39] D. K. Berlo, *The Process of Communication: An Introduction to Theory and Practice*. Austin, TX: Holt, Rinehart and Winston, 1960.

[40] D. Chandler, "The transmission model of communication." Online short paper at `http://visual-memory.co.uk/daniel/Documents/short/trans.html`, 1994. Accessed on 31 July 2017.

[41] W. Schramm, *How communication works*. 1954. (Reprint in) *Mass Media and Society* by Alan Wells, ch. 3, pp. 51–63. Santa Barbara, CA: Greenwood Publishing Group, 1997.

[42] D. C. Barnlund, "A transactional model of communication," in *Communication Theory* (C. D. Mortensen, ed.), ch. 4, pp. 47–57, Piscataway, NJ: Transaction Publishers, 2007.

[43] J. A. Anderson, *Communication Theory: Epistemological Foundations*. New York, NY: Guilford Press, 1996.

[44] J. Cassell, J. Sullivan, S. Prevost, and E. F. Churchill, *Embodied Conversational Agents*. Cambridge, MA: MIT Press, 2000.

[45] L. Sun, A. Shoulson, P. Huang, N. Nelson, W. Qin, A. Nenkova, and N. I. Badler, "Animating synthetic dyadic conversations with variations based on context and agent attributes," *Computer Animation and Virtual Worlds*, vol. 9, pp. 17–32, 2012.

[46] J. van Oijen and F. Dignum, "Agent communication for believable human-like interactions between virtual characters," in *Cognitive Agents for Virtual Environments (CAVE): First International Workshop, Held at AAMAS 2012, Valencia, Spain, June 4, 2012, Revised Selected Papers* (F. Dignum, C. Brom, K. Hindriks, M. Beer, and D. Richards, eds.), (Heidelberg, Germany), pp. 37–54, Springer-Verlag, 2013.

[47] S. I. Park, F. Quek, and Y. Cao, "Simulating and animating social dynamics: Embedding small pedestrian groups in crowds," *Computer Animation and Virtual Worlds*, vol. 24, pp. 155–164, 2013.

[48] C. M. Henein and T. White, "Microscopic information processing and communication in crowd dynamics," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 21, pp. 4636–4653, 2010.

[49] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Pearson Education, Inc., 2003.

[50] H. S. Nwana, "Software agents: An overview," *The Knowledge Engineering Review*, vol. 11, no. 03, pp. 205–244, 1996.

[51] T. Finin, R. Fritzson, D. McKay, and R. McEntire, "KQML as an agent communication language," in *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM)*, (Gaithersburg, MD), pp. 456–463, ACM, 1994.

[52] J. R. Searle, *Speech Acts: An Essay in the Philosophy of Language*, vol. 626. Cambridge, UK: Cambridge University Press, 1969.

[53] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE – a FIPA-compliant agent framework," in *Proceedings of the Fourth International Conference on Practical Application of Intelligent Agents and Multi-Agents (PAAM)*, (London, UK), pp. 97–108, 1999.

[54] B. B. Mandelbrot, "How long is the coast of Britain," *Science*, vol. 156, no. 3775, pp. 636–638, 1967.

[55] P. M. Torrens, A. Nara, X. Li, H. Zhu, W. A. Griffin, and S. B. Brown, "An extensible simulation environment and movement metrics for testing walking behavior in agent-based models," *Computers, Environment and Urban Systems*, vol. 36, no. 1, pp. 1–17, 2012.

[56] A. Nara and P. M. Torrens, "Spatial and temporal analysis of pedestrian egress behavior and efficiency," in *Proceedings of the 15th ACM International Symposium on Advances in Geographic Information Systems (GIS)*, (New York, NY, USA), pp. 59:1–59:4, ACM, 2007.

[57] K. Kullu, U. Güdükbay, and D. Manocha, "ACMICS: an agent communication model for interacting crowd simulation," *Autonomous Agents and Multi-Agent Systems*, May 2017.

[58] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA: MIT Press, 2005.

[59] G. Snook, "Simplified 3D movement and pathfinding using navigation meshes," in *Game Programming Gems* (M. DeLoura, ed.), pp. 288–304, Boston, MA: Charles River Media, 2000.

[60] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[61] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[62] A. Stevenson, ed., *Oxford Dictionary of English*. Oxford, UK: Oxford University Press, 3 ed., 2010.

[63] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*. Upper Saddle River, NJ: Pearson Education, Inc., 2011.

[64] K. Kullu and U. Güdükbay, "A layered communication model for agents in virtual crowds," in *Proceedings of 27th International Conference on Computer Animation and Social Agents (CASA 2014), Short Papers*, (Houston, USA), May 2014.

[65] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation.* Pearson, 3 ed., 2007.

[66] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee, "Group behavior from video: A data-driven approach to crowd simulation," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pp. 109–118, ACM SIGGRAPH/Eurographics Association, 2007.

[67] B. Zhan, D. Monekosso, P. Remagnino, S. Velastin, and L.-Q. Xu, "Crowd analysis: a survey," *Machine Vision and Applications*, vol. 19, pp. 345–357, 2008.

[68] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert, "Data-driven crowd analysis in videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (Barcelona, Spain), pp. 1235–1242, 2011.

[69] A. Bera and D. Manocha, "Realtime multilevel crowd tracking using reciprocal velocity obstacles," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)*, (Stockholm, Sweden), pp. 4164–4169, IEEE, 2014.

[70] J. Sime, "Crowd psychology and engineering," *Safety Science*, vol. 21, no. 1, pp. 1–14, 1995.

[71] Unity Technologies, "Unity-game engine." `http://unity3d.com/`. Accessed on 15 July 2017.

[72] X. Pan, *Computational Modeling of Human and Social Behaviors for Emergency Egress Analysis.* PhD thesis, The Department of Civil and Environmental Engineering, Standford University, 2006.

[73] Integrated Environmental Solutions Ltd., "Simulex." https://www.iesve.com/software/ve-for-engineers/module/Simulex/480. Accessed on 31 July 2017.

[74] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLoS ONE*, vol. 5, no. 4, pp. 1–7, 2010.