A Decision Network Framework for the Behavioral Animation of Virtual Humans

by

Qinxin Yu

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

A Decision Network Framework for the Behavioral Animation of Virtual Humans

Qinxin Yu

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2007

We introduce a novel framework for advanced behavioral modeling in virtual humans that features the first application of decision network techniques to computer graphics and addresses the challenging open problem of simulating the complex interactions of real people in urban settings. Based on hierarchical decision networks, our behavioral modeling framework handles uncertainty, interprets perceptual information, incorporates personality traits, and facilitates the autonomous control of behavior. Combining probability, decision, and graph theories, our framework yields autonomous human characters that can make nontrivial interpretations and arrive at intelligent decisions that depend on multiple considerations.

Unlike prior work in so-called "crowd simulation", we develop complex autonomous individuals that, in addition to motor and perceptual components, include broad behavioral repertoires that are much more challenging to model. In particular, our self-animating pedestrians can independently assess the interrelationships among all the relevant factors to make rational decisions in the presence of uncertainty.

Within our framework, we develop an emergency response behavior model, which enables virtual pedestrians to respond to an emergency situation in a variety of ways. We also develop a behavior model for establishing partnering relationships, an acquaintance behavior model enabling two characters to greet one another, and a collision avoidance model, all of which take into consideration personality traits, internal factors, and the

interpretation of intentions.

Our virtual human simulator, which includes the aforementioned behavioral models, can automatically animate pedestrians in a large urban environment that interact with each other in a realistic manner. We demonstrate the potency of our decision network framework in several behavioral animation scenarios involving interactions between autonomous pedestrians, including an elaborate, automatically-generated emergency response animation.

Acknowledgements

First and foremost, I would like to thank my advisor Professor Demetri Terzopoulos. His passion and enthusiasm for computer graphics inspired me and has had a profound influence on me. I have learned a great deal from Demetri not only on research, but also on what constitutes a great mentor. His breadth of knowledge, his openmindedness and his expertise in technical writing have benefited me greatly. I have always come out more excited about my work after each discussion with him. This thesis would not have been possible without his expert guidance and unfailing encouragement.

I am honoured to have had Professors Eugene Fiume, Fahiem Bacchus, and Karan Singh on my Ph.D. committee. It has been a privilege to hear their thoughts on my research. Their insightful comments helped to improve my dissertation greatly and I thank them for helping me put my thesis into perspective.

My special thanks go to Professor Norman Badler for serving as my external examiner. His valuable critique helped to make my thesis complete. I greatly appreciate his time and effort.

I would also like to thank Professors Michiel van de Panne, Alejo Hausner, John Danahy, and James Stewart for their assistance and support in the earlier stages of my graduate program.

Dr. Wei Shao kindly provided his software system for use in my research. His generosity and patient answers to my questions quickened and smoothened what would otherwise have been a time consuming and hairy coding process.

Linda Chow and Julie Weedmark helped me to sort out the tedious details in arranging my defense, and I would also like to thank Marina Haloulos for her assistance throughout the years.

The wonderful people at the DGP lab provided a warm and rich environment for conducting my graduate study. I really thank them for the great time I have had at the UofT. I would like especially to thank Bowen Hui for introducing me to the concept of decision networks, which became the foundation for my framework in the thesis; Faisal Qureshi, Joe Laszlo, Patrick Coleman, and Michael Neff for valuable discussions; Sam Hasinoff, Hanieh Bastani, and Mike Daum for being voice actors for my animation clips—they are so natural that they should consider careers in Hollywood; John Hancock, Anastasia Bezerianos, and Alan Rosenthal for providing excellent support as lab administrators; and the rest of the gang for making my time at the UofT so memorable and full of joy.

My deep appreciation goes to Cathy Yansen, Jingrui Zhang, Irene Fung, Maciej Kalisiak, Michael McGuffin, Gonzalo Ramos, Michael Wu, Marge Coahran, Petros Faloutsos, Victor Ng, Chris Trendall, Nick Torkos, Naomi Friedlander, Nigel Morris, Jimmy Talbot, Jeff Tupper, Glenn Tsang, Yan Wang, Naiqi Weng, Hai Wang, and Meng Sun for their friendship and lively chats.

Special thanks go to my fellow Falun Gong practitioners for having nothing to do with my thesis, but everything to do with the great time I had outside of school.

Finally, an enormous thank you to my parents and my husband, whose unconditional love and support made it possible for me to get to where I am today.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Virtual human modeling has numerous applications ranging from entertainment (interactive games and movie special effects), to architecture and urban planning, to human factors analysis, to archeology, to clinical diagnosis. Creating autonomous agents that are able to reproduce human-like behaviors has been a challenging objective, which has attracted many researchers from different disciplines. From modeling the motor system, human behaviors, to modeling facial expression, speech recognition and generation, etc., each of these important components has spawned significant research of its own.

The work presented in this dissertation addresses advanced behavioral systems for virtual humans. By a behavioral system, we mean the level of decision-making that allows the characters to react appropriately to the perceived environment, including other virtual humans. We focus on the action selection component of a behavioral system; that is, we are concerned with simulating how humans decide what to do at any given time. In particular, we introduce a *decision network framework* for specifying human behaviors that is easy to define, easy to modify, scalable, and mimics the way humans make decisions; for instance, in the automatically generated "emergency response" animation depicted in Figure 1.1.

Decision networks are a decision theoretic generalization of Bayesian networks [Pearl

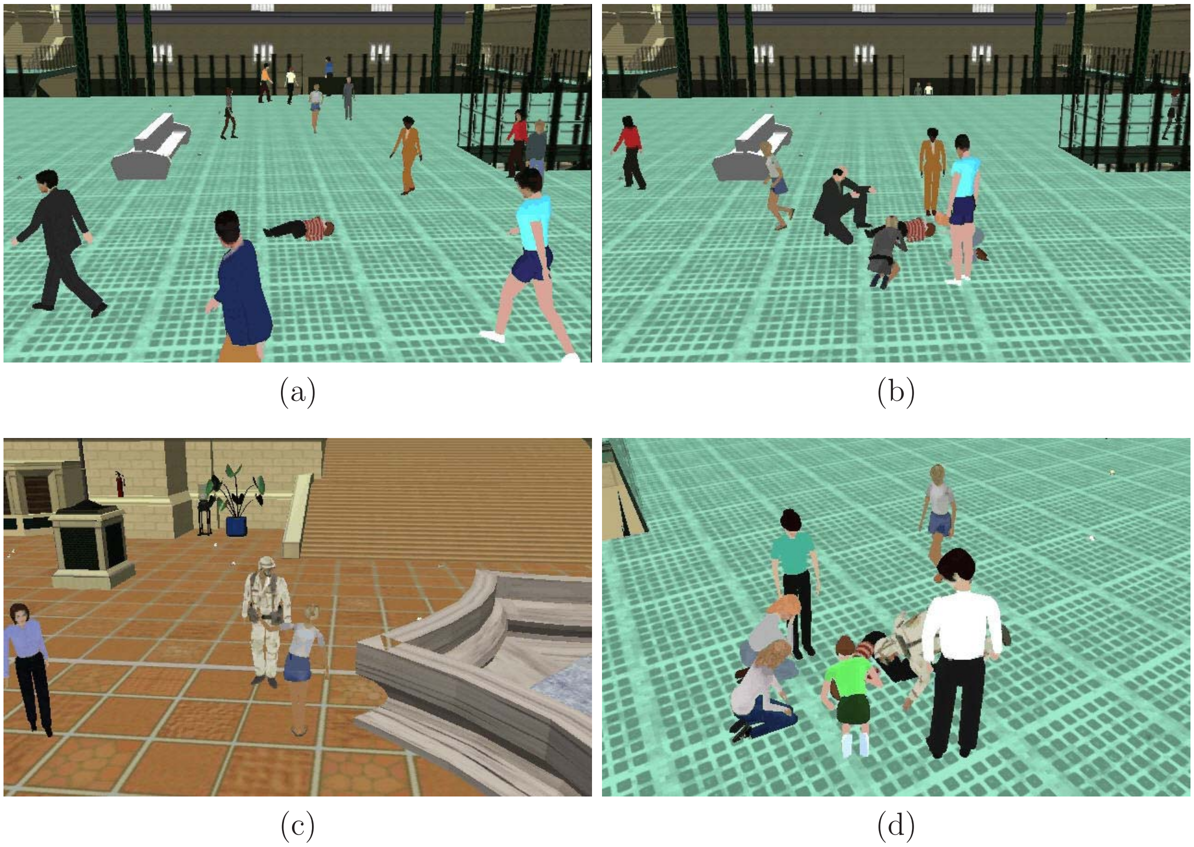


Figure 1.1: Still frames from an emergency response animation. (a) A female character passed out on the floor in the concourse of a virtual train station. (b) Commuters gather around to attend to the patient; a woman runs for help. (c) The woman finds a law enforcement officer in the main waiting room of the station. (d) The officer attends to the patient.

1988], also known as probabilistic graphical models, which combine probability theory and graph theory to capture uncertain knowledge in a natural and efficient way. The decision network is a powerful tool for modeling decision making under uncertainty. It provides an elegant mathematical structure for modeling complicated relationships among random variables and an intuitive visualization of these relationships via the graphical structure, thus facilitating comprehension and debugging. Furthermore, the modularity of a decision network facilitates the intuitive subdivision of a complex behavior model into manageable components. We use decision networks as the basis of our framework which addresses several tough challenges in behavior modeling, which we will discuss next.

1.1 The Behavior Modeling Challenge

Human behavior modeling and simulation is a challenging problem. The framework that we develop in this thesis aims to handle uncertainties, facilitate the simulation of character interactions, incorporate the simulation of perceptual focus of attention, and facilitate the specification of behavior rules.

Uncertainty: When people know enough facts about their environment, logic will in principle enable them to derive plans that are guaranteed to work. Unfortunately the real world is fraught with uncertainty and we almost never have access to the whole truth about our environment. One source of uncertainty is the limitation in our perception, not only because we often have incomplete knowledge of the environment, but also because of our inability to understand all of its complexities and how they affect one another. For example, we may not be certain if it is going to rain, but we may assess that the probability of rain at 70%, and base our decision to carry an umbrella on this assessment. The role of uncertainties becomes even more apparent during interaction between people, as one person's decision is often based on the other person's reactions, which cannot fully be known in advance. We observe the world and take action based on our observations. However, our perception of the world may or may not be an accurate account of the state of our surroundings, and our decisions are made based on our assessment of our environment in consideration of our intentions, goals, personality, etc. In fact, we often face the need to make an unclear choice among a set of unclear alternatives. In reality, uncertainty exists everywhere, so in order to truly simulate human behaviors realistically, uncertainty is an important factor that must be addressed.

(Mis)Interpretation and Interaction: We do not act directly on the stimuli that come to the attention of our sensory organs, but on our mind's interpretation of these stimuli. That is, we make decisions based on our impression or interpretation of the world around us, not necessarily on the objective existence of things in our surroundings. This becomes more important when we think about the interaction among characters. In the real world, when people interact with each other, there is a perpetual process of interpreting each other's intention, we either react to it or take initiative and start some action. The interpretation in this case is a conclusion we draw about the other person's intention, emotion, etc., based on observed cues about the other person. This interpretation may or may not be accurate, but it will influence our reaction. In the real world, it is impossible to know the exact feelings and intentions of the other person, and interesting behavior patterns arise as a result of misinterpretation. For example, occasionally when trying to avoid collisions, two opposing pedestrians will move in the same direction almost simultaneously, then again to the opposite direction at the same time, which results in a temporary deadlock, until one party decides to stop and allow the other to proceed to one side. For the sake of realism, it is important to model such misinterpretations.

Decision Specification: Decisions are often made among competing goals. The greater the number of possible goals and actions we take into consideration, the harder it becomes to define rules for deciding what to do. In simulating human behaviors, we naturally try to mimic real human behavior, albeit in a highly simplified way. We humans have an established common sense about how people normally do things, and we have gained some understanding through our experiences and observations. Hence, an important factor to consider when designing a behavior system is how easily and intuitively to make the transition from the knowledge we have gained about how people behave, to the representations that the system uses to determine how characters make decisions.

Controllability: When an animator designs a character, he/she may want to have a certain level of control over the character's actions. There is always a tradeoff between autonomy and directability. For example, it was reported in the popular press that when making the motion picture trilogy "The Lord of the Rings", during early tests in creating the battle scenes using the Massive system (www.massivesoftware.com), some of the soldier characters "decided" to flee the battle rather than to fight fiercely as the director intended them to do. This element of surprise can be interesting, but it can also be frustrating when the animator has strict requirements for the characters. Although we want to simulate autonomous characters, we also want to provide the animator with an intuitive mechanism for directing these characters by adjusting how they make decisions.

Extensibility: Another factor to consider is how to make the behavior system easily extensible. Often we need to make adjustments to the specifications, whether to add in another attribute or to add a new behavior routine to extend the character's functionality. The system needs to be flexible and self-consistent so that modifications can be easily incorporated without having to make large changes to the specifications of existing behavior routines. The system should also facilitate the testing of new specifications.

1.2 Contributions

Unlike so-called "crowd simulation", which has been intensively studied by the computer animation research community, we concentrate on simulating individual behaviors. In conventional crowd simulation, the characters in a crowd follow one another according to predefined procedures that are based on simple rules. This thesis focuses on simulating autonomous individuals whose behaviors are much more complex to model. The decision process can no longer be accomplished with a centralized system. It requires the characters to independently sort through the complicated interrelationships among all relevant factors. Our more sophisticated model affords the characters the capability of performing a variety of activities, including acting as a group. We have implemented a system, which simulates the appearance, locomotion, perception, behavior and cognition of pedestrians in an indoor urban environment, although the behavior framework that we develop can be applied more generally to other human simulation scenarios.

Our primary contribution in this thesis is the introduction and development of a new framework for simulating human behaviors. Our framework features the first application of decision networks to computer graphics. More specifically, it addresses the challenges in human behavior modeling identified in the previous section. Hence, it offers the following advantages over prior behavioral animation methods:

1. Our framework enables graphical characters to reason with preferences and to make rational decisions in the presence of uncertainties. Uncertainty has largely been ignored in previous human behavior models.

2. Our framework facilitates the encoding of common sense knowledge into the behavior specifications, which determine how decisions are made. In particular, complex interrelationships among factors in an advanced behavior model can be represented systematically and explicitly in the decision network's graphical representation, which simplifies the specification, communication, and debugging of the models.

3. The modularity of a decision network enables the intuitive division of a complex behavior model. This simplifies the extension and fine tuning of a behavior model.

4. Unlike prior techniques, our decision network framework offers a systematic and powerful interpretation mechanism, which facilitates the simulation of interactions by enabling characters to interpret one another's intentions.

5. Our framework models the effects of personality and internal factors on decision making, thus enhancing the realism of human behavior simulation. We have also incorporated the simulation of perceptual focus of attention in our virtual pedestrian model.

We introduce four specific behavior models that are implemented within our framework:

1. *Emergency response behavior model:* This high-level model enables the characters to respond to an emergency situation in a variety of ways. Some may come over to investigate, some may fetch help and some may simply avoid the situation, depending on how serious they think the incident is, their own personalities and conditions, and what they perceive others are doing about it.

2. *Partnering behavior model:* This model enables the characters to form partnerships with each other. The establishment of a partnering relationship requires consent from both participants, where each character is making its decision based on not only its own desires but also on its interpretation of the potential partner's willingness to cooperate.

3. *Acquaintance model:* This model enables two characters who encounter each other to decide how to greet one another. The decision depends on each character's attraction and feeling towards the acquaintance, as well as its interpretation of the acquaintance's intention.

4. *Collision avoidance behavior model:* In this novel low-level model, in addition to analyzing the potential obstacle's walking direction and speed in relation to the character's own goal, the character also takes consideration of its interpreted obstacle's avoidance strategy.

Using our virtual pedestrian system, which includes the above models, we create several new automatically-generated animations of virtual humans that demonstrate the effectiveness of our framework.

1.3 Thesis Overview

The remainder of this dissertation is organized as follows:

Chapter 2 reviews related prior work. In addition to a general literature review of prior research on behavior modeling and human behavior modeling in particular, previous uses of Bayesian networks in other areas of computer graphics are also covered.

Chapter 3 provides background information on Bayesian networks, and it formulates decision networks and their inference processes.

Chapter 4 presents a rationale for choosing decision networks as the basis of our framework and provides an overview of the framework.

Chapter 5 presents an overview of our virtual human model and the environment model in which the virtual pedestrians are situated.

The main technical portion of the thesis, Chapters 6 and 7, develops the behavior models that we have implemented within our decision network framework, which include (high-level) emergency response behavior, partnering behavior, acquaintance behavior, and (low-level) collision avoidance behavior. Chapter 6, overviews these behavior models, while the network structures and parameter settings are detailed in Chapter 7, which also illustrates the modeling of personality and internal factors affecting decisions.

Chapter 8 presents and discusses selected animation results.

Finally, Chapter 9 draws conclusions and proposes promising avenues for future work.

Chapter 2

Related Work

In this chapter, we review prior work in the field of behavior modeling, which is the focus of this thesis. Behavioral animation focuses on modeling the behavior of the character, rather than its shape or physical properties. Reynolds [1987] introduced the term and the concept of behavioral animation. He used a particle system to simulate realistic polarized, aggregate motion, such as that of flocks, herds and schools. The model simulates the behavior of each bird-like "boid" independently. The boids try to group together and at the same time avoid collision with each other and other objects in the environment.

Since then, behavioral animation has attracted many researchers. In this thesis, we will focus on the action selection aspect of behavioral modeling. That is, we are concerned with how the synthetic actors decide what to do at any given time. There is a tradeoff between directability and autonomy. When a character is scriptable, it will do what the animator specifies, but then the animator must specify everything, which is inappropriate for spontaneous interaction. By contrast, when a character is autonomous, it saves work for the animator and is good for interaction, but the characters may be hard to control and may not always do what the animator desires. Three categories of techniques are described in the following sections: script based methods, reactive/motivational behavior methods, and cognitive modeling, in order of increasing autonomy.

2.1 Script-Based Methods

In the "Improv project" [Perlin and Goldberg 1996], the authors use scripts to control the decisions made by the character's mind. The scripts are organized into groups, and at any time, only one script within a group can be running. Generally, a group will contain scripts specifying alternative modes in which an actor can be at some level of abstraction. Improv also enables the user to handle nondeterministic behaviors by associating weights with actions or scripts, and creating decision rules to specify how relevant information influences the weight associated with each choice. This approach gives the animator the control over the character, and is capable of handling interactive applications, but the animator must give detailed specifications, including timing information for each action, etc. Therefore, this approach becomes impractical for producing complex autonomous behaviors or a large variety of behaviors. Unlike our approach, script-based methods are hard to specify and cannot easily be extended or modified.

2.2 Reactive/Motivational Behavior Methods

Another approach is to make reactive and motivational decisions based on the character's internal state and its perception of external stimuli. Drawing on ideas from the field of ethology, Tu and Terzopoulos [1994] implement an ethological behavior model and an intention generator for their artificial fishes. At each time step, the intention generator issues an intention based on the fish's habits, mental state and incoming perceptual information. Appropriate behavior routines arranged in a loose hierarchical structure are then invoked to satisfy this intention.

Blumberg and Galyean [1995] present a related computational model of action selection. They organize activities in loosely overlapping hierarchies with more general activities at the top and more specific activities at the leaves. A value is computed for each activity based on external factors and internal factors. All the constituent parts of a behavior are accessible at runtime, and the user can direct the autonomous animated creatures at multiple levels. In their recent work [Blumberg et al. 2002], the authors have used reinforcement learning to train an autonomous animated dog.

Bates et al. [1994] set up a plan memory where one or more plans are stored for each goal. These plans are ordered or unordered collections of subgoals and actions that can be used to accomplish the invoking goal. When there are multiple plans for a given goal, one will be chosen at execution time based on perception, emotional state, behavior features and other aspects of internal state. When one plan fails, the system will backtrack and choose another to accomplish the invoking goal. A priority number is assigned to each instance of a goal, which is used to determine which goal to follow next. Their text based system combines emotion with behaviors and it modeled some interesting cat behaviors.

Our approach is also reactive and motivational, but the behavioral modeling framework that we develop in this thesis enables our characters to make inferences about their surroundings, a capability which was lacking in previous methods. In addition, we focus on the ability to handle uncertainties, which is not addressed in previous techniques.

2.3 Cognitive Modeling

Previous methods focus on how to make characters react appropriately to perceived environmental stimuli, whereas the work of Funge et al. [1999] concentrated on modeling what a character knows, how that knowledge is acquired, and how it can be used to plan actions over longer time frames. The authors develop a cognitive modeling language, CML, and use the logic-based situation calculus formalism from artificial intelligence to perform the reasoning. Their model is deliberative rather than reactive. It successfully modeled herding behaviors and intelligent evasion behaviors for a merman in shark infested waters, and it was also used for camera control. However, as human reasoning is exceedingly complex, it is unclear how well this method will be able to model the full range of human cognition. Rather than focusing on planning, our technique focuses on allowing the characters to make rational decisions in action selection in the presence of uncertainties.

2.4 Virtual Human Behavior Modeling

In the previous subsections, we reviewed past work on modeling animal behaviors. Human behaviors are much more complex and are affected by many factors. Virtual human modeling is a broad topic that includes many aspects. This thesis work will mainly address the behavior modeling of virtual humans. More specifically, it will concentrate on characters that perform autonomously in a virtual world. Hence, the research will concentrate more on the character's interaction and autonomy in the virtual environment and less on dialog or natural verbal communication.

Researchers at the University of Pennsylvania developed the Jack software, which is used worldwide for human figure animation and human factors analysis. They have designed a Parameterized Action Representation (PAR) [Badler et al. 2002], which includes specifications for low-level animation concepts, such as kinematics and dynamics, as well as descriptions for primitive or complex actions. Selection of actions in the system (PARSYS) is based on the applicability conditions (Boolean expression), preparatory specifications (conditions-and-actions sequence) and termination conditions (Boolean expression) specified in the corresponding PAR structure.

Silverman et al. [2006] made use of human performance moderator functions in their PMFserv architecture to simulate virtual humans. It uses three value trees, the Goal Tree, the Standards Tree, and the Preference Tree (referred to as GSP trees) and their weights (utilities) to represent the character's knowledge. Using an OCC emotion model, the characters select action(s) based on the utilities in their GSP trees.

Devillers et al. [2002] developed a programming environment including Hierarchical

Parallel Transition Systems (HPTS), a behavioral programming language, and a scenario language for behavioral animation. HPTS is a formalism that can be used to address hierarchical concurrent behaviors. It has a reactive and a cognitive model, which can be viewed as a multi-agent system in which agents are organized as a hierarchy of state machines. The behavioral programming language fully implements the HPTS formalism and is used to give behavior descriptions. The scenario language describes scenarios in a hierarchical manner and schedules them at simulation time.

Noser and Thalmann [1996] use a timed, conditional, stochastic, parametric and environmentally sensitive L-system to describe and animate autonomous actors which are completely defined by production rules. Musse and Thalmann [1997] simulated crowd behavior by modeling the relationship between autonomous virtual humans in a crowd and the global effect that emergent from local rules. Each virtual human is given a set of parameters, with rules defining how these parameter values can change during the simulation, and there are also rules defining how these parameters drive the group's behavior. Ulicny and Thalmann [2001] simulated crowd behavior for interactive virtual environments using behavior rules and hierarchical finite state machines.

As part of the Virtual Theater project, Doyle [1999] embeds necessary knowledge including appropriate responses and emotions into the virtual environment through annotation, so that the agents can extract it and use it for guidance as they decide what to do. This makes the agents appear expert on the local content of the world and can react appropriately when the world changes around them.

Johnson's [1995] WavesWorld equips each character with goal, sensor and skill, and then uses an action selection network to connect these components together. The action selection algorithm is an extension of Maes' algorithm with competence modules [Maes 1989].

Bates et al. built "broad agents", which aim at producing a broad range of abilities that are integrated into a coherent system rather than to perfect any individual aspect of the agents. As part of the Oz project [1995], Reilly and Bates simulate a negotiation behavior over trading cards, by including as many responses as possible and the ability to interrupt behaviors with other behaviors. They also tried to incorporate emotion and personality into the behavior, the emotions generated depend on several factors including the importance of success, the importance of not failing, a likelihood of success function, a likelihood of failure function, a function to determine who is responsible if the goal succeeds, and a function to determine who is responsible if the goal fails. The behaviors themselves are specified with rules and condition statements.

VRaptor developed by Shawver [1997] is a VR system for situational training. The virtual world is populated by actors under the control of trainer-defined scripts, while the trainees are represented by avatars. Each actor uses a modified version of the NYU KPL language interpreter to govern its behaviors. A hostage rescue scenario was enacted with this system.

Shao and Terzopoulos [2005] developed an autonomous pedestrian model in the setting of a virtual train station. Their virtual pedestrians combine behavioral and cognitive modeling, and are capable of performing a variety of activities. Their behavior model is based on that of Tu and Terzopoulos [1994].

Most behavioral systems for virtual human modeling are rule based, but as will be explained in Section 4.1, rule based systems are not appropriate for handling uncertainties. Neither can they easily make inferences based on observations. Our approach aims to address both issues, which are vital for enhancing the realism of autonomous characters.

Some researchers have studied the social aspect of virtual humans. Cassell et al. [1999] explored autonomy that comes from underlying models of social and linguistic intelligence for their embodied conversational characters. These characters respond to interactional cues such as gestures and audio input based on a set of states and are able to participate in face-to-face conversation with a human. The system used a uniform knowledge representation format throughout. Guye-Vuillieme and Thalmann [2001] used a role-based organization to give their characters the ability to perform some social reasoning. For example, their characters are able to greet one another differently based on their gender, etc.

Some researchers have addressed the simulation of attention and gaze behavior. Chopra-Khullar and Badler [2001] proposed a computational framework for generating visual attending behavior of virtual human characters, which controls eye and head motions. They drew upon empirical and qualitative observations from psychology and implemented behaviors as parallel, executing finite state machines. Peters and Sullivan [2003] implemented a system to automatically generate bottom-up visual attention behavior for virtual humans. Their system comprises a visual sensing component to perceive the virtual world, an attention component for early processing of perceived stimuli, a memory component to store previously sensed data, and a gaze generation component for the generation of resultant behaviors.

There are many efforts on behavior modeling in Robotics research as well, such as Mataric's [1997] work on behavior-based control and Brook's [1999] work on humanoids robots. Lebeltel et al. [2004] established a new method to program robots based on Bayesian inference and learning. They have tested their system with a two-wheeled mobile robot. Breazeal [2003] developed an expressive anthropomorphic robot that is capable of generating emotions in response to visual and auditory stimuli provided by humans who are interacting with it, and expressing that emotion through face and body posture. Due to robot mobility limitations, behavior modeling in this area is mainly concerned with robotic movements; thus far, it has not addressed the emulation of realistic high-level human behaviors, which is the theme of this dissertation.

2.5 Interpersonal Understanding

Humans have the ability to understand others as intentional agents. As there is no objective way to ascertain other people's consciousness or to assess their motivations and desires, people can only guess at these things when they interact with each other, trying to interpret what they know, think, or feel. There can be quite a few aspects to how one interprets the actions and intentions of other people. Baron-Cohen [1995] approached the question with the notion of intentionality, which may be primarily inferred on the basis of motion. It can be categorized along an intensity dimension from "carrying out routine but generally goal-directed activities" to "desperately wanting to achieve a certain goal." Blakemore and Decety [2001] studied how humans have an inherent tendency to infer other people's intentions from their actions.

The detection of eye direction is another cue that can be utilized to infer what another person is likely to do. Baron-Cohen [1995] found that infants are very sensitive to eye direction, particularly for eye-contact detection. In practice, eye-contact detection appears to trigger a variety of responses. For example, Brooks [1999] and his students built a robot "Cog" with video-camera eyes. Brooks reported[1] that a woman who knew quite a bit about Cog's construction found herself involuntarily flattered that the robot followed her with its eyes as she crossed the room. In another situation, she might have felt threatened. Peters [2005] used Bayesian networks to calculate the likelihood that another character wants to engage in a conversation based on a character's current direction of attention, short term history of attention, and any directed gestures made. A character that shows a high interest in the subject is perceived to be likely to engage in further conversation. The agent's behavior is guided by a finite state machine.

Shared attention refers to the re-orientation or re-allocation of attention to a target because it is the object of another person's attention. It plays a critical role in many

[1] In a November 1997 talk at the Society for Literature and Science in Pittsburgh.

social activities, such as learning and drawing inferences about other people's current and future activities. Gaze following is one of its manifestations.

One's emotion can be communicated through words, gestures, body language, facial expressions, etc. Multiple emotions play a role in regulating cooperative behaviors. For example, in a social setting, one's emotions may indicate a willingness or unwillingness to cooperate. Baron-Cohen et al. [1996] inferred emotions from facial expressions.

The research on interpersonal understanding of mental states serves as the basis for determining how the characters can draw interpretations about others based on limited cues. In our current system, gaze direction is used as an important cue in making inferences.

2.6 Decision Theory

Almost everything that a human does involves decisions. Decision making is the reasoning process which determines a final choice among alternatives. The choice made can be an action or an opinion. The particular type of human activities we are interested in are goal-directed behaviors in the presence of options.

Decision theory is a theory about decision making. As developed by philosophers, economists, and mathematicians for over three hundred years, decision theory has developed many powerful ideas and techniques that have had a major influence across the cognitive and social sciences. Decision theory can be classified as normative or descriptive. Normative decision theory is concerned with identifying the best decision to make, i.e., how decisions should be made, whereas a descriptive decision theory attempts to describe how decisions are actually made.

Most decisions take time and can be divided into phases or stages. As part of his motivation for the French constitution of 1793, Condorcet [1793] gave the first general theory of the stages of a decision process. He divided the decision process into three stages. In the first stage, one "discusses the principles that will serve as the basis for decision in a general issue; one examines the various aspects of this issue and the consequences of different ways to make the decision." In the second stage, the decision is reduced to a choice between a manageable set of alternatives. In the third stage, the actual choice between these alternatives is made.

The starting point for modern decision theory is generally taken to be Dewey's [1910] decomposition of the decision process into five stages. Simon [1960] modified it to comprise three principal phases: "finding occasions for making a decision; finding possible courses of action; and choosing among courses of action." The focus of this thesis is on the last phase of this decision process decomposition. Dewey's and Simon's models are both sequential as they have divided decision processes into parts arranged in the same sequence. A criticism for it is that a more realistic model should allow various parts of a decision process to be executed in different orders for different decisions. Mintzberg, Raisinghani, and Theoret [1976] subsequently proposed a non-sequential model, in which they used the same three major phases as Simon, but divided each phase into two routines and gave a circular instead of a linear relationship between these phases and routines.

Utility was first formalized by Leon Walras, and was improved by Ramsey [1931], and later by von Newmann and Morgenstern [1944]. Utility is a measure of the happiness or satisfaction gained from performing a certain action. It is the mathematical representation of an individual's preferences over alternatives. Von Nuemann and Morgenstern extended this concept to establish the modern theory of choice under uncertainty—the theory of expected utility. In the context of expected utility, the utility of an agent facing uncertainty is calculated by considering utility in each possible state and constructing a weighted average. The weights are the agent's estimates of the probability of each state.

Decision theory combines probability theory with utility theory. It provides a formal and complete framework for decisions made under uncertainty. A good decision should identify an alternative that is at least as good as any other alternative actions. The orthodox approach to decision theory is to define rational decision makers as those that always maximize expected utility. Bayesianism assumes that choices between alternative courses of action under uncertainty can be decided by expected utility [Ramsey 1931; de Finetti 1937; Savage 1954]. There are also approaches that adopt non-probabilistic notions of belief, different notions of worth, or criteria other than expected utility maximization [Doyle and Thomason 1999]. For example, many contributions have been made in qualitative decision theory [Doyle 1980; Bacchus and Grove 1996; Doyle and Thomason 1999] to handle decisions where the decision maker is thinking in categories ("likely", "unlikely", etc.) instead of exact probabilities, generic preferences, and generic goals.

2.7 Bayesian Networks in Computer Graphics

Ball and Breese [2000] encoded emotions and personality with Bayesian networks. In their architecture for a character-based user interface, they used a Bayesian network to estimate the likelihood of specific body postures and gestures for different personality types and emotions in generating responses to the user's interaction. Their emphasis is on conversational agents with speech recognition and generation. Kshirsagar [2002] also used a Bayesian network to model personality and mood for their chat application. Egges et al. [2003] used a Bayesian network to model mood based on the agent's current mood, expected mood and changed mood for their conversational agent. In the computer vision system of Oliver et al. [2000], they used a Bayesian approach to detect and recognize different human behaviors and interactions in a visual surveillance task. Their work concentrated on computer vision and machine learning rather than behavior modeling.

To our knowledge, the work closest to our own includes the work of [Hy et al. 2004] on using Bayesian networks to simulate simple behaviors for a first person shooter game character. In their paper, the authors used a Bayesian network to specify finite state machine-like behavior selection, and to learn by imitating a human player. However they did not use Decision networks, nor did they simulate any human interactions. To the best of our knowledge, no prior behavior system has been built in the field of computer graphics that uses decision networks as its fundamental base.

Chapter 3

Background on Bayesian and Decision Networks

3.1 Bayesian Networks

Bayesian networks, also known as "belief networks" or "probabilistic graphical models", combine probability and graph theory to capture uncertain knowledge in a natural and efficient way [Pearl 1988]. Bayesian networks are directed acyclic graphs in which each node represents a random variable, arcs represent direct causal influence between the linked variables, and the strengths of these influences are quantified by conditional probabilities. Probabilistic beliefs about the connection strength are updated automatically as new information becomes available.

As a mathematical basis for Bayesian networks, Bayes' theorem states that

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)}, \tag{3.1}$$

where $P(X|Y)$ is the posterior probability, or the probability of X given the observation Y, where $P(X)$ is the prior probability and $P(Y|X)$ is the likelihood, or the probability of the observed information Y given a particular event X. The quantity $P(Y)$ is the unconditional probability of the data, also called the normalizing constant or prior

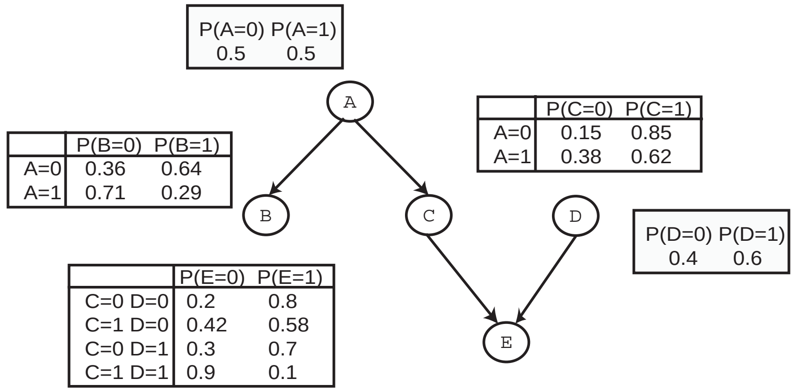


Figure 3.1: A sample Bayesian network

predictive distribution.

The topology of the network; i.e., the set of nodes and links, specifies the conditional independence relationships among the variables. The topology along with the conditional probability distribution defined for each variable given its parents, suffice to represent the dependencies among the random variables and to specify (implicitly) the full joint distribution for all the variables.

Figure 3.1 shows a sample Bayesian network with five discrete variables. Each variable has two possible values. For example, variable C has one parent and one child node, whereas variable E has two parents and zero children. The joint probability of a Bayesian network is defined as the product of local conditional probabilities:

$$P(X_1, X_2, \dots, X_n) = \int_{i=1}^{n} P(X_i|\text{Parents}(X_i)). \tag{3.2}$$

So, for example, the joint distribution corresponding to the network of Figure 3.1 is given by:

$$P(A, B, C, D, E) = P(A)P(B|A)P(C|A)P(D)P(E|C, D). \tag{3.3}$$

The semantics of a Bayesian network is that each node is conditionally independent from its non-descendants given its parents. More generally, two disjoint sets of nodes X and Y are conditionally independent given E, if E d-separates X and Y, where d-separation is defined as follows:

Definition. d-separation: A set of nodes E d-separates two other sets of nodes X and Y if every path from a node in X to a node in Y is blocked given E [Korb and Nicholson 2004].

A path and a blocked path are defined as follows:

Definition. Path (Undirected Path): A path between two sets of nodes X and Y is any sequence of nodes between a member of X and a member of Y such that every adjacent pair of nodes is connected by an arc (regardless of direction) and no node appears in the sequence twice.

Definition. Blocked path: A path is blocked, given a set of nodes E, if there is a node Z on the path for which at least one of three conditions holds: Z is in E and Z has one arc on the path leading in and one arc out (chain). Z is in E and Z has both path arcs leading out (common cause). Neither Z nor any descendant of Z is in E, and both path arcs lead in to Z (common effect).

3.1.1 Construction

A Bayesian network can be regarded as a representation of the joint probability distribution, and it gives a complete description of the domain. This view is helpful in understanding how to construct networks. Every entry in the full joint probability distribution can be computed based on information provided in the network. The value of any generic entry in the joint distribution is given by:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(X_i)), \tag{3.4}$$

where $\text{Parents}(X_i)$ denotes the specific values of the variables in the parent nodes of X_i. Thus, each entry in the joint distribution can be represented by the product of associated elements in the conditional probability tables (CPTs) in the Bayesian network.

The joint distribution can be rewritten in terms of a conditional probability using the product rule $P(a \wedge b) = P(a \mid b)P(b)$ as

$$P(x_1, \ldots, x_n) = P(x_n \mid x_{n-1}, \ldots, x_1)P(x_{n-1}, \ldots, x_1).$$

As we repeat this process of reducing each conjunctive probability to a conditional probability and a smaller conjunction, we end up with

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_{i-1}, \ldots, x_1). \tag{3.5}$$

This identity is known as the chain rule, and it holds true for any set of random variables.

Comparing equation 3.5 with equation 3.4, it can be shown that the specification of the joint distribution is equivalent to the following equation for every variable X_i in the network:

$$P(X_i \mid X_{i-1}, \ldots, X_1) = P(X_i \mid \text{Parents}(X_i)),$$

provided that $\text{Parents}(X_i) \subseteq \{X_{i-1}, \ldots, X_1\}$. This means that in order for the Bayesian network to give a correct representation for the domain, each node must be conditionally independent of its predecessors in the node ordering, given its parents. Hence, in constructing a network, we need to choose parents for each node according to this property. An intuitive way to think about this is that the parents of node X_i should contain all those nodes in $X_1, \ldots, X_{i-1}$ that directly influence X_i. A correct construction of the network will have the network topology to reflect the direct influences with the appropriate set of parents, and allow the minimum number of nodes and links necessary to represent the domain. The correct order to add nodes is to "add the 'root causes' first, then the variables they influence, and so on, until we reach the 'leaves', which have no direct causal influence on the other variables" [Russell and Norvig 2003].

3.1.2 Inference

When data of interest cannot be observed directly, then an inference procedure is typically required to determine them. This involves calculating marginal probabilities conditional on the observed data using Bayes' theorem, which is diagrammatically equivalent to reversing one or more of the Bayesian network arrows.

The probability of any event A can be computed by conditioning it on any set of exhaustive and mutually exclusive events $B_i, i = 1, 2, \ldots, n$:

$$P(A) = \int_i P(A|B_i)P(B_i) \tag{3.6}$$

With a Bayesian network, we can compute any conditional probability; hence, we can answer all possible inference queries by marginalization (summing out over irrelevant variables). However, the number of variables involved in computing the conditional probabilities is exponential, and the complexity for general querying of a Bayesian network is complete for #P [Roth 1993]. Much research has been done in finding efficient inference methods, such as variable elimination [Aji and McEliece 2000; Kschischang et al. 2001], the junction tree algorithm [Cowell et al. 1999; Huang and Darwiche 1994], and a variety of approximation methods [Jordan et al. 1998; Jaakkola and Jordan 1999]. These algorithms run in exponential-time in the worst case, but when the treewidth w of the network graph is small, they run in time and space $n^{O(1)}2^{O(w)}$, where n is the number of nodes in the network [Bacchus et al. 2003]. Most of the networks involved in our work have low treewidth, hence inferencing does not pose a problem.

3.2 Decision Networks

It is often said that "Decision Theory = Probability Theory + Utility Theory". Probability theory describes what an agent should believe based on evidence gathered about the current state, while utility theory describes what an agent wants. Decision theory combines the other two theories together to describe what an agent should do. A decision network (also called an inference diagram (ID)) [Howard and Matheson 1981; Shachter 1990] extends a Bayesian network by adding actions and utilities, where utility node(s) will have as parents all the attributes (random variables) on which they depend as well as action node(s), since the utility depends both on the state of the world and the action we perform. According to Russell and Norvig [2003], "Utility is a function that maps from states to real numbers." The utility function captures the agent's preferences between world states and assigns a number to express the desirability of a state.

A decision network encapsulates information about the agent's current state, possible actions, the state that will result from the agent's action, and the utility of that state. There are three types of nodes in a decision network, namely the chance nodes (ovals), which represent random variables and each current-state chance node can be part of a Bayesian network; the decision nodes (rectangles), which represent the choice of actions the agent can make; and the utility nodes (diamonds), which represent the agent's utility function. In this thesis, we will use a simplified form [Russell and Norvig 2003] in which the chance nodes that describe the outcome state are omitted and the utility nodes are connected directly to the current-state nodes and the decision node. Hence, the utility node will represent the expected utility ($EU(A|E)$) associated with each action (A) given the evidence (E) as defined by

$$EU(A|E) = \sum_i P(\text{result}_i(A)|Do(A), E)U(\text{result}_i(A)), \tag{3.7}$$

where result(A) represents possible outcome states of action A.

An action-utility table is associated with each utility node to specify the corresponding expected utility values and reflect any changes in circumstances which affect the outcome state, as well as changes in the weight associated with the utility functions. Decision networks can be used to compute the optimal (sequence of) action(s) to perform so as to maximize expected utility, although this is computationally intractable for all but the smallest problems.

A decision network can have multiple action and utility nodes; however, it does not have a temporal component like a dynamic Bayesian network does. The dynamic decision network is essentially an augmented dynamic Bayesian network with utility nodes and decision nodes for actions. It provides a general, concise, more structured representation for large, partially observable Markov Decision Processes [Tatman and Shachter 1990].

3.3 Netica Software

In implementing our decision network framework, there is no need to reinvent the wheel. We have made use of a commercial Bayesian and decision network software package supplied by Norsys Software Corp. called *Netica* (www.norsys.com). The role played by Netica in this thesis is no different than the role played by, say, Matlab and Maya in many CG research projects.

Netica serves simply as a software tool. We use the Netica API to construct, compile, perform inference and evaluate the networks. The Netica software provides a user interface for drawing the networks, which gives a graphical view of the network. We use it to generate belief-bar diagrams, which will be used in Chapter 7 to illustrate our network structure and evaluation results. The relationships between variables can be entered as individual probabilities in the form of equations or learned from data files. In our system, we specify them as individual probabilities.

Netica uses the junction tree algorithm; hence, the constructed networks must first be compiled into a junction tree of cliques for fast inference. When we are certain that a variable is in one of its states, this observation can be entered into the network as hard evidence. When we are not completely sure about particular findings, they can be entered as likelihood. The likelihood is a vector containing one probability for each state of the corresponding node. Each component of this vector is between 0 and 1 inclusive, but they are not required to sum to 1. Once the findings are entered, the network will be updated.

Compiled nets can be queried to obtain posterior probabilities for any node. Decision networks are solved using the junction tree algorithm to find optimal decisions and conditional plans. In our system, we use Netica API commands to carry out relevant operations.

Chapter 4

Decision Network Framework

A decision network extends a Bayesian network by including utilities and actions. It enables rational decisions based on what the agent believes and what it wants, whereas a logical agent would not be able to handle uncertainty combined with conflicting goals [Russell and Norvig 2003]. Our framework exploits decision networks as the basic representation. In our application, not every decision need take all sensory and internal factors into consideration. Hence, to avoid the potential intractability of computing large decision networks, we build our behavioral models as hierarchical collections of smaller decision networks. At the lower level, a separate, smaller decision network structure is implemented for each decision item, while at the higher level(s) the decision network structure at each node represents how a decision is made based on results from its children nodes.

4.1 Rationale for Choosing Decision Networks

Current behavioral modeling techniques are largely rule-based. Rule-based systems have three desirable properties, namely, locality, detachment, and truth-functionality. With locality, given a rule of the form $A \mapsto B$, whenever evidence A is given, one can conclude B without worrying about any other rules. This avoids complex interactions. With detachment, propositions are detached from their justifications and can be used regardless of how they were derived. With truth-functionality, the truth of a complex sentence can be computed from the truth of its subsentences. However, these properties are not appropriate for reasoning in the presence of uncertainty [Russell and Norvig 2003]. They include degrees of belief, which has serious implications when applied to uncertain problems. The truth-functionality property includes rules for intercausal reasoning, and forward and backward chaining, which cause serious conflicts when dealing with uncertainties, as propagation of evidence is not tracked. For example, if we have the rule $HighUV \mapsto BringUmbrella$ (allowing one to compute the belief in bringing an umbrella for protection as a function of the belief in the rule and the belief in the UV index being high on the day), and the rule $BringUmbrella \mapsto Rain$. In a truth-functional system, when $HighUV$ is true, chaining forward through the rules, it increases the belief that one will bring an umbrella, which in turn increases the belief that it is raining. However, in reality, the fact that UV is high already explains away the need to bring umbrella; hence, it should reduce the belief in it being a rainy day.

As we were searching for a framework that is capable of dealing with uncertainties, we considered some alternatives. Neural networks [McCulloch and Pitts 1943; Bishop 1995; Fausett 1994] have been successfully applied to pattern recognition problems, and they are good at representing complicated, nonlinear functions. However, the approach relies on the concept of training an initially random network to correctly map inputs to outputs. In our application, it is difficult to obtain sufficient data to train neural networks. Furthermore, neural networks do not give explicit knowledge representation in easily interpretable forms. The model is implicit, hidden in the network structure and optimized weights; i.e., one would have little or no idea how it reaches a conclusion, which makes it harder to identify the cause of an inappropriate decision outcome and tune the network. Compared with neural networks, Bayesian networks also have the advantage that an expert can provide knowledge in the form of causal structures.

Another alternative is fuzzy logic[Zadeh 1988], which really handles degree of truth, not uncertainty. As a truth-functional approach, it is not capable of taking into account the correlations or anticorrelations among component propositions[Russell and Norvig 2003]. Due to vagueness, it can be harder to attribute an outcome to the inputs that generated it.

In comparison, Bayesian networks have several advantages in general. An attractive feature of the Bayesian network is that it provides an elegant mathematical structure for modeling complicated relationships among random variables while employing a relatively simple visualization of these relationships. This makes it easier to comprehend and debug. In addition, Bayesian networks are easier to maintain given the modular representation of uncertain knowledge, and they are more readily extensible. Given a hierarchical structure, the addition of a new piece of evidence may require only the addition of a fairly small number of probabilities, in addition to a small number of edges in the graph. Bayesian networks are direct representations of the world, not of the reasoning processes. Unlike rule-based systems and neural networks, the arrows in Bayesian network diagrams represent causal connections and not the flow of information during reasoning. Reasoning processes can operate on Bayesian networks by propagating information in any direction; hence, they are capable both of making predictions based on some evidence and of explaining some observations that we make about the world.

4.2 Addressing the Identified Issues

We will now explain how our proposed framework addresses the issues identified in Section 1.1.

Uncertainty: The uncertainties associated with various variables of interest are represented by the probability distributions encoded in the decision network. These probability distributions are updated as time progresses. In essence, the probabilities in the networks are measures of our belief given the current state of knowledge. Things that are known for certain by the character, such as its internal states are entered as evidence into the network. Preferences and priorities in certain circumstances are encoded in the utility functions of the decision network.

(Mis)Interpretation: Bayesian and decision networks are direct models of the world. The reasoning processes can operate on these networks by propagating probabilistic information in any direction; hence, they are capable both of making predictions based on some evidence and of explaining some observation we make about the world. The inference that the character makes about its environment and other characters are inferences made by probabilistic (Bayesian) reasoning processes in the network. The probability distributions within the network can be adjusted to increase or reduce the chance of making an accurate interpretation.

Decision Specification: A decision network encodes events, represented by nodal variables, and causal relationships between them represented by directed edges. The network encodes information about the character's current state, possible actions, the state that will result from the character's action, and the utility of the resulting state. This is particularly desirable for defining behavior rules, since the decision making process can be viewed as a causal relationship—certain circumstances and intentions lead to certain decisions. The decision network provides an elegant mathematical structure for modeling complicated causal relationships among events while providing an informative visualization of these relationships. Thinking about causal relationships is more intuitive. We simply need to consider the relevant factors, express their relative importance by assigning corresponding conditional probabilities, and specify the desirability of possible courses of action in the utility functions. Regarding the concern about where to obtain prior probabilities, in our application this mainly relies on the character designer's commonsense knowledge. These priors need not be exact to be useful. They can be adjusted and fine-tuned further if necessary.

Controllability: The use of decision networks provides a convenient way to control how the character makes decisions. Adjusting the conditional probabilities and the utility functions will influence how the decision gets made. Another way to exert control is to adjust internal parameters, which will be monitored by the network during the simulation in order to make inferences and assessments. For example, if boldness is a deciding factor, increasing the boldness parameter on the fly, will transform the character from being a timid character to a bold one, and its subsequent decisions will reflect this change.

Extensibility: Decision networks are easier to maintain given the modular representation of uncertain knowledge. By virtue of the hierarchical structure, the addition of new evidence usually requires the addition of only a fairly small number of probabilities and edges in the graph. Since it is easy to map a decision network structure into corresponding code, debugging and modification becomes a matter of adjusting the directed graph associated with a decision network and the corresponding utility values and/or conditional probabilities.

Chapter 5

System Overview

5.1 Virtual Human Model

To evaluate our framework, we have implemented a virtual human model based on the autonomous pedestrian software that was developed by Shao and Terzopoulos [2005], including their virtual environment model of the original Pennsylvania Train Station in New York City. Figure 5.1 illustrates the architecture of the virtual pedestrian character. Like real pedestrians, the synthetic humans sense the virtual environment around them, interpret the sensory stimuli, make decisions based on their perceptual interpretations, and act in accordance with their decisions. The major advance in our human model relative to that in [Shao and Terzopoulos 2005] is our behavior submodel, in which we exploit decision networks to simulate the interactions between multiple pedestrians and to model the effect of different personalities on their decisions, as will be detailed in the next section.

5.1.1 Motor System

Following [Shao and Terzopoulos 2005], we use Boston Dynamics Inc.'s *DI-Guy* API, which provides a variety of textured geometric models of humans and motion libraries

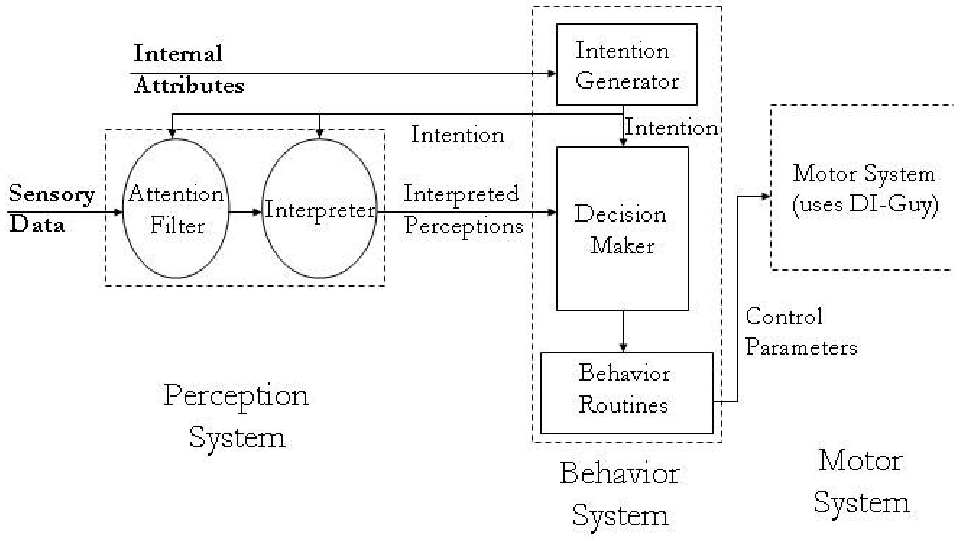


Figure 5.1: Autonomous virtual human model.

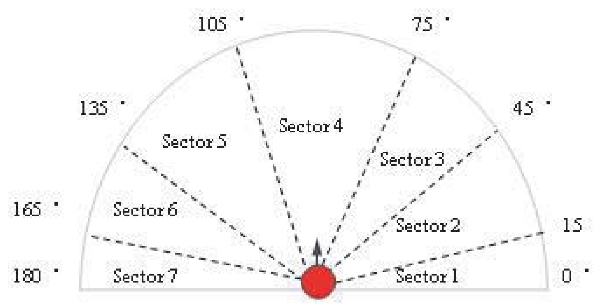


Figure 5.2: Field of view of a virtual pedestrian.

capable of generating some basic human motions such as walking and running. Unfortunately, DI-Guy characters have certain limitations both in their appearance and in their motor skills. For example, they are incapable of quickly changing speeds like real humans, which limits the quality of their collision avoidance behavior. Unfortunately, this detracts from the realism of our animations. However, our higher-level behavior model is designed to be independent of the lower-level motor system.

5.1.2 Perception System

The perception system of the virtual human is responsible for sensing and interpreting the world. Each pedestrian has a 180° field of view, which is divided into seven sectors (Figure 5.2). An occupancy index value, which combines a dynamic obstacle occupancy index and a static obstacle occupancy index, is computed for each sector to indicate how crowded it is. The dynamic obstacle occupancy index is calculated based on how far away the dynamic obstacle is, and if the obstacle is moving in the same or opposite direction of the character. The static obstacle occupancy index mainly depends on how far away the obstacle is.

People are aware of things around them only if they attend to them. When something falls into our attention, we notice it and then process or interpret it. Attention can shift rapidly, switching from one object of interest to another. Two classes of mechanisms exist to control attention: top-down or goal-driven and bottom-up or stimulus-driven [Egeth and Yantis 1997]. Some visual attention shifts are initiated voluntarily on the basis of a particular goal (hence the term "goal-driven control of attention"), as when we look for a water fountain. Other shifts appear to be triggered reflexively in response to a sudden stimulus event (hence the term "stimulus-driven control of attention"), as when our attention is grabbed by a cat that suddenly jumps onto the sidewalk.

Through focus of attention, we decide what to attend to and what to ignore. We used the aforementioned two mechanisms to simulate focus of attention. In the absence of a sudden stimulus, the character will focus on the objects of interest to fulfill its goal; otherwise, a sudden stimulus has priority. The attention focus in turn determines the gaze direction for the character.

Objects that enter into the pedestrian's attention are what it actually perceives at the moment. Hence, further processing is conducted on these objects to effect perception.

5.1.3 Behavior System

As is shown in Figure 5.1, the behavior system mediates between the perception and motor systems. The behavior system of our virtual human model comprises a set of behavior routines, an intention generator, and a decision maker that is based on decision networks. Chapters 6 and 7 present the details of our decision network based behavior models for virtual pedestrians.

The limitations of the motor system pose a challenge to the behavior system, however. Although the decisions made by our decision network framework are general, there is a limited variety of motions available from the DI-Guy motion library that can be used to represent corresponding actions. We have designed behavior routines to meaningfully couple the decisions made to the DI-Guy motor system.

In order for one virtual human to interpret the behavior of another, it has to make observations. Since facial expression and gestures are highly constrained in the motor system, currently there are only a limited set of cues upon which our virtual humans can base their observations. These include change in direction, change in speed, gaze direction, and orientation. Change in speed is an especially unreliable visual cue as the DI-Guy motor system lacks the ability to change speed very quickly.

Personality

Another important aspect of our behavior system is a model of individual personality. In psychology, *personality* is considered a collection of emotional, behavioral, and thought patterns unique to a person that remains consistent over time, and is what differentiates one individual from another. According to the *Diagnostic and Statistical Manual* of the American Psychiatric Association [American Psychiatric Association 1994], personality traits are “prominent aspects of personality that are exhibited in a wide range of important social and personal contexts;” i.e., that a person’s personality influences his/her behavior. To simulate human behaviors realistically, one must consider the effects of personality in decision making.

After decades of research, the scientific community is reaching consensus on a general taxonomy of personality traits, the "Big Five" model [Goldberg 1981], which posits that human personality can be described along five broad dimensions, regardless of language or culture. These fives dimensions, Extroversion, Agreeableness, Conscientiousness, Neuroticism, and Openness, were derived from the analysis of empirical data collected from thousands of answers to hundreds of questions. Extroversion refers to the degree to which a person can tolerate sensory simulation from people and situations. High extroversion makes one tend to be social, friendlier, and talkative, whereas low extroversion makes one tend to be private, serious, and skeptical. Agreeableness concerns cooperation and social harmony. An agreeable person is considerate, friendly, generous, helpful and willing to compromise his/her interests in consideration of the interests of others. Disagreeable individuals tend to be unconcerned with the well-being of others, and place self-interest above getting along with others. Conscientiousness refers to the way people control, regulate and direct their impulses. Conscientious persons tend to show self-discipline, act dutifully and aim for achievement. Neuroticism refers to the tendency to experience negative feelings. It is related to the ability to handle pressure and stress. Those scoring high on neuroticism tend to respond with more intense emotion than normal and are more likely to interpret ordinary situations as threatening. Openness to experience distinguishes imaginative, creative people from down-to-earth, conventional people. A more open person is more prone to boredom and curiosity, and tends to be more artistic.

Three of the personality traits described in the Big Five model are especially relevant to our behavior system—extroversion, agreeableness, and conscientiousness. The extroversion trait is considered in the acquaintance model to decide how a character should greet to another known character, as well as in the partnering model, since it affects how the character interacts with others. The agreeableness trait is modeled in the emergency response model as it concerns with how caring the character is about others. The conscientiousness trait is considered when the character tries to decide if it should end its emergency response behavior, as a patient character may not mind hanging around longer as much as an impatient one would. The other two traits in the Big Five model are not directly relevant to the behaviors we model, and hence are not explicitly modeled in our current system.

Internal Factors

In addition to personalities, *internal factors* also affect how the characters behave, as together with external stimuli, they describe the full circumstance under which the decision has to be made. For example, whether the character is in a hurry or how friendly it feels towards a particular character are among the internal factors that affect the character's decision.

Given the same external stimuli, characters with different personality and internal factors may arrive at different decisions. Our framework is able to simulate the influence these factors have on behaviors.

Memory

Our autonomous characters can perform many activities. In the train station setting, for example, a character can stop to buy a ticket, or stop to talk with a friend on his way towards the platform, or watch a performance while waiting for his train. At any given time, the character may have multiple goals in mind. While carrying out the sequence of goals, the character may be interrupted with some sudden event with which he must deal before continuing his original plan, such as to avoid a collision, or to encounter a friend, etc. The character must remember the sequence of tasks it wants to do as well as sudden intermediate goals that attract the character's attention. For this purpose, we implement a memory in the character's behavior system as was done in Shao's system [Shao and Terzopoulos 2005].

We use a stack as the data structure for implementing memory. At any given time, the item at the top of the stack list is being processed. This restriction deprives the character from being able to perform multiple tasks in nondeterministic order. However, this limitation has little noticeable effect on the character's observable behaviors. Instead, the stack implementation has the advantage of taking constant time for operations regardless of stack size. This property is particularly desirable for simulating large numbers of characters.

With their memory, the characters can remember their tasks, and hence maintain persistence in their behaviors. Intermediate tasks can be added without causing the characters to forget what they originally planned to do. The stack memory operates as follows: At any given time, the top goal on the stack is processed. If the goal is accomplished or expires[1], it is removed from the stack. If there is a sudden event, which requires the character's immediate attention, such as collision avoidance or response to an emergency situation, the corresponding task is added to the top of the memory stack as an intermediate task with an appropriate expiration time. Hence, after this intermediate task is completed, the original goal will be on top of the stack again and will be processed, allowing the character to resume its course of action before the interruption.

When the top goal requires multiple sub-tasks to accomplish, a plan is made for it based on the current situation, which consists of a sequence of sub-tasks, each with appropriate expiration times. Instead of replacing the original goal with this new sub-task sequence, the first sub-task in the list is added to the memory stack, and the original goal is updated with parameters indicating the goal decomposition status. The reason for this is that the goal decomposition is often a dynamic process that depends on environmental conditions at the time. If we simply replace the original goal with the decomposed sub-tasks, it limits the character to completing the task with the original decomposition done

[1]Goals are given an expiration time limit, as there are some tasks that take long and cannot be completed, they should be given up after being pursued for a certain period of time

at the beginning, even though during the process of completing the task, the real-time situation may render it no longer an optimal solution. Thus, only the first sub-task is added to the memory stack and, by the time it is completed, the original goal will be processed again, and the goal decomposition can be performed based on the then current situation, hence improving the character's ability to adapt to the changing environment.

The characters also have a short term memory to remember the observations that they have made regarding the objects of their attention from the previous time step. These include the gaze direction, walking direction, and walking speed of the character in its focus, and the character's own interpretation of the other character's intentions. These data in memory assist a character in judging if the other character had changed gaze direction, walking direction, or speed, and reassessing its intentions.

5.2 Environment Model

The virtual environment provides a synthetic world in which the virtual humans are situated and can interact. The environment model used by our system is the virtual train station model described in Shao [2005]. This section provides a brief overview of this model.

The virtual environment is a reconstructed 3D model of the original Pennsylvania Train Station of New York City. Figure 5.3 shows a cutaway side view of the rendered train station geometric model with the long shopping arcade at the left, the main waiting room at the center, and the two-level concourse over the train tracks at the right. Geometrically, the station is a 3D space whose dimensions are 200m (length) x 150m (width) x 20m (height), featuring a total of over 500 objects, including levels of floors, stairs, walls, doorways, big columns, ticket booths, platforms, train tracks, lamps, a fountain, benches, vending machines, tables, sporadic trash on the floors, etc. Figure 5.4 shows two views of the populated virtual Penn Station.

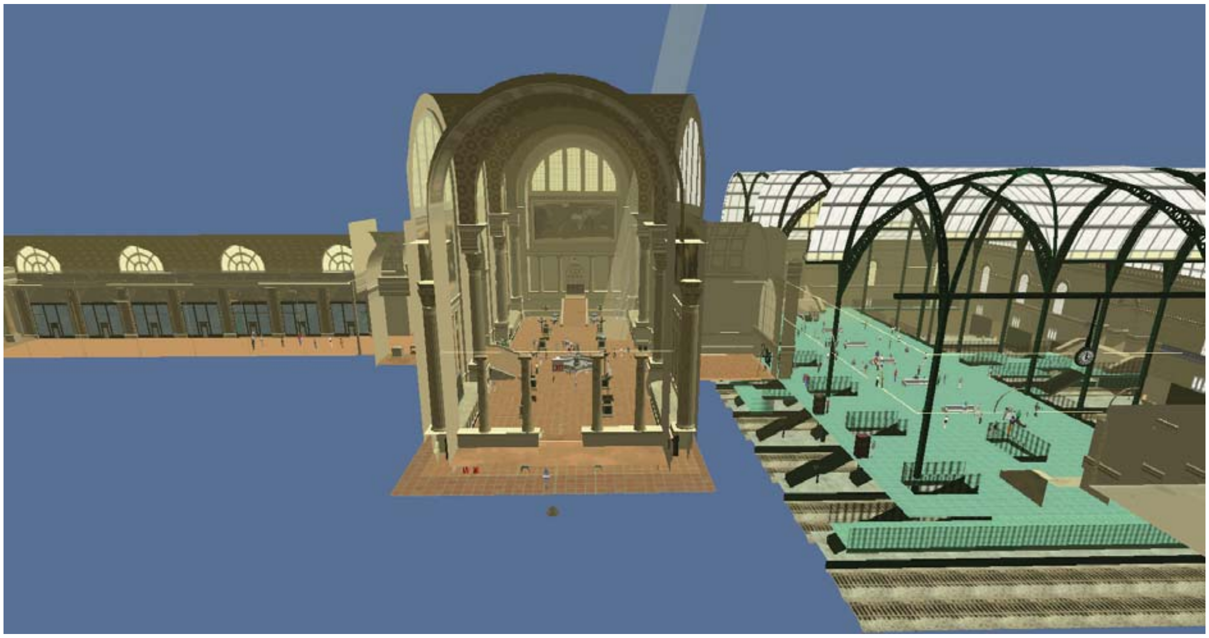


Figure 5.3: A cutaway side view of the Penn Station model. The main waiting room is at the center, the arcade at the left, and the concourses on the right with the train tracks underneath them.

The virtual environment is represented by a hierarchical collection of maps. At the top level, a topological map is used to represent the topological relations between different parts of the virtual world. Its nodes correspond to environmental regions, where each region is a bounded volume in 3D space, including all the objects inside that volume. It is assumed that the 3D space of a region can be effectively mapped onto one or more horizontal planes. The nodes also include precomputed *path-to-via* information, which indicates the path of minimal cost that goes from one point to another via this region. Hence an entry in the *path-to-via* information $V(A, T)$ answers the question of to which region should the character go, and what cost should it expect if it is currently at A and wants to reach T. The edges in the topological map represent accessibility between regions. Linked within each node of the topological map are perception maps and path maps.

The locations of stationary objects and mobile objects are represented in two separate perception maps. For stationary objects, the region is rasterized into a uniform grid,

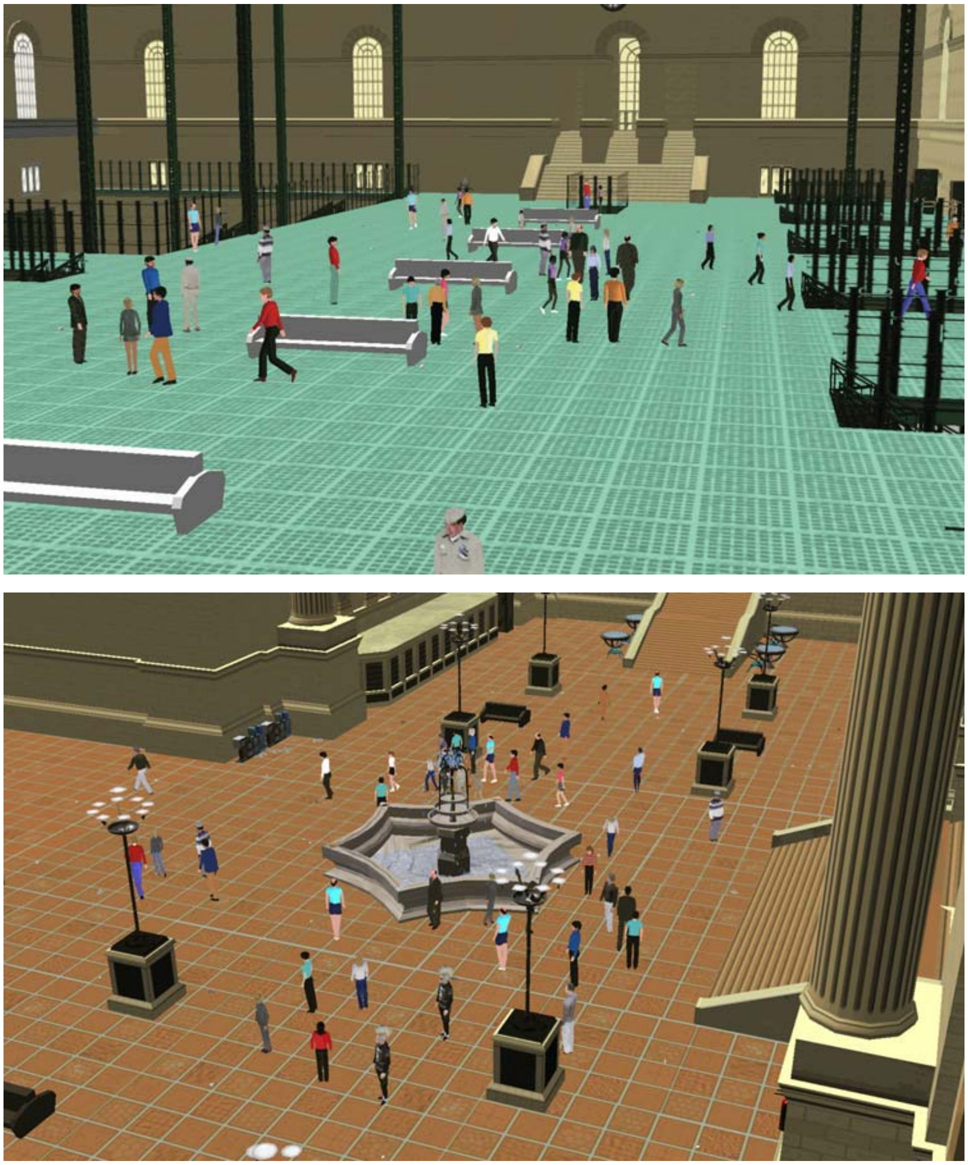


Figure 5.4: Autonomous pedestrians in the virtual Penn Station. Concourse area (top) and the waiting room (bottom).

where each cell corresponds to a small area of the region, and identifiers of all the objects that occupy that small area are stored in the corresponding cell. The visual sensing query simply shoots out a fan of rays and interrogates the grid cells along each ray for their associated object information. The length of these line segments reflects the desired perceptual range, while the density reflects the desired perceptual acuity. In our simulation, the typical cell size of perception maps for stationary object perception is 0.2

to 0.3 meters.

For mobile objects, a 2D grid map is used in which each cell stores and updates the identifiers of all the characters currently within its area. Queries about finding pedestrians near a given character within its sensing range can be answered efficiently by examining this map.

Two types of path maps, namely the grid maps and the quadtree maps, are used and they are differentiated by their data structures. One can always find the shortest path on a grid map; hence, grid maps are used whenever a detailed path is needed and they are also used for short-range path planning. The quadtree maps support the execution of several variants of the A* graph search algorithm, and are used for global and long-range path planning.

At the lowest level of the environment hierarchy are specialized objects such as the Ground, the Seat, and the Waiting-line, etc. They efficiently provide query information that higher level perception maps cannot.

Chapter 6

Behavioral Modeling Using Decision Networks

We have applied our decision network framework to the development of interaction models between virtual humans, guided by our commonsense knowledge of how real humans behave in similar circumstances. In the following sections, we present four behavior models that we have developed—specifically emergency response behavior, partnering behavior, acquaintance behavior, and collision avoidance behavior—ordering them in top-down fashion, from high to low level. This chapter presents an overview of these behaviors models, while Chapter 7 will present their complete details. The emergency response behavior will serve as a concrete example for us to describe in detail how the networks are constructed, how the network parameters are adjusted and, subsequently, how we can obtain various possible decisions automatically from them and even extend them.

6.1 Emergency Response Behavior

Our highest-level and most elaborate behavior model is the emergency response behavior, which simulates how people might respond to an emergency situation in a public space like the train station.

6.1.1 Network Construction

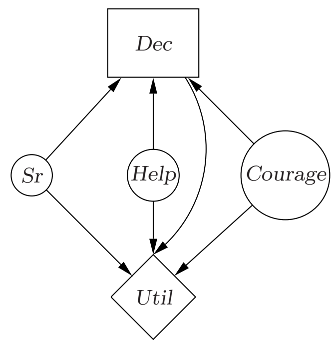
Regardless of the methodology that one uses to design a behavior model, be it rule-based or probabilistic, it is natural for the designer to think in terms of what factors cause what effects and eventually lead to a decision underlying action selection. For human character animation, it often suffices to use common sense and imagine how one would consider various factors to make the appropriate decision when encountering the scenario in real life. Decision networks provide a natural mapping of this thought process to the behavior model representation.

Although the order of variables in the decision network structure is important to ensure a correct model, in practice a proper ordering can be readily determined by asserting causal relationships among variables in the domain. The correct ordering is achieved by simply instantiating arcs from cause variables to their immediate effects, and one can start this process by considering the root cause. As soon as we have determined the variables representing the factors we would like to consider and the causal relationship among them figured out, we effectively have designed the network topology. It then remains to complete the network by filling in the associated probability or utility values. Hence, the process of constructing the decision networks starts with a consideration of what choice of actions we want our characters to have and what decision(s) they need to make. We then need to consider what factors influence such decisions and what interpretations must be made that affect the decisions. Once we establish the causal relationships, the character's preferences, and how the character should weigh each influencing factor, we can construct the networks by naturally following our intuitions. The Netica software provides a graphical user interface for defining the networks, enabling the user easily to create nodes, set their corresponding probabilities, insert links, and make modifications. The GUI also offers a graphical view of the network, so that one can clearly see its structure. The constructed networks can then be saved in a Netica recognizable format, which can be read by the Netica API into our animation system software.

In the current context, our goal is to model how people respond to an emergency situation. We approach the task by considering what possible reactions we might see in people encountering such a situation. When someone collapses in a public place, those who witness the event may react in different ways. Some people may approach to investigate, while others may prefer to avoid the situation and carry on with what they were doing. Of the people who do decide to approach the scene, the more concerned will run over, while the less concerned will walk over. These are the three initial reactions we wish to model, so they will be the three possible actions from which our virtual pedestrian character can select.

Next, we consider the factors that affect the action selection decision. Three main factors come to mind. The first is how serious the character thinks the situation is. The second is how much the character wants to help others. A third contributing factor is how courageous the character is. A timid character may lack the courage to confront a potentially unpleasant scene at close range, hence shy away from an emergency situation. Having selected these three factors as those that contribute to the action selection decision, we simply use a variable to represent each factor, and create a chance node for each variable. Each factor is represented by a binary variable, as both the seriousness measure and whether the character is helpful can take the value of either *yes* or *no*, while the variable representing the character's courage can take on values of either *strong* or *weak*. The action selection decision is represented with a decision node, whose value can take on one of the three possible actions. We also need to add a utility node to indicate the character's preferences over the three input factors on the decision. Figure 6.1 shows the decision network comprising these nodes. The links between nodes indicate causal relationships. Since the three factors contribute to the decision, we have added an arc to link from each of these three factors to the decision node. An arc is also added to link from each of the three factors and the decision node to the utility node, as all are inputs for the utility table to be specified at this node.

P(Sr)		P(Help)		P(Courage)	
yes	no	yes	no	s	w
sr	1.0-sr	he	1.0-he	n	1.0-n

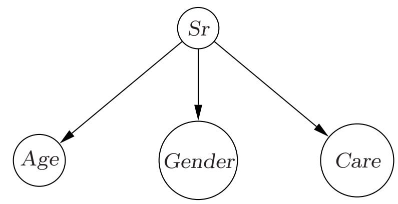


Sr=yes				Sr=no			
Help	Courage	Dec	Util	Help	Courage	Dec	Util
yes	s	ig	-10	yes	s	ig	-5
yes	s	run	8	yes	s	run	5
yes	s	obs	6	yes	s	obs	3
yes	w	ig	-8	yes	w	ig	3
yes	w	run	6	yes	w	run	2
yes	w	obs	7	yes	w	obs	4
no	s	ig	1	no	s	ig	5
no	s	run	3	no	s	run	1
no	s	obs	5	no	s	obs	2
no	w	ig	3	no	w	ig	6
no	w	run	1	no	w	run	0
no	w	obs	2	no	w	obs	2

Figure 6.1: Network to determine how to respond to the emergency. In the CPT, Sr: how serious the situation is, Help: how much the character wants to help others, Courage: how courageous the character is, which can take on the values of strong (s) or weak (w). The parameters he and n are the character's corresponding personalities. The character's decision Dec can take on the values of ig (ignoring the incident), run (running over to check out the situation), or obs (walk over to observe).

To complete the network, we must define the prior probabilities for the three chance nodes. The seriousness of the situation must be assessed, so its value should be set based on the interpretation result. To keep the network structure modular and hierarchical, the interpretation result can be obtained from a separate network. The prior probabilities for the help and the courage nodes are set directly based on the character's internal factors and personality. For example, if the character's willingness to help others is 0.6 in a range from 0.0 to 1.0, the priors for the corresponding node Help are set to 0.6 for the *yes* state

P(Sr)	
yes	no
0.5	0.5



	P(Age)			P(Gender)		P(Care)	
Sr	y	p	o	male	female	yes	no
yes	0.3	0.2	0.5	0.3	0.7	0.9	0.1
no	0.3	0.5	0.2	0.7	0.3	0.1	0.9

Figure 6.2: Network to assess the seriousness of the situation. In the CPTs, the symbol Sr denotes the perceived seriousness of the situation. For the node representing the patient's age, the states y, p, o correspond to young, prime age, and old, respectively.

and 0.4 for the *no* state. Similarly the priors for the Courage node are set based on how courageous the character is. The utility table entries are set based on our intuition about the character's preference over the three possible actions given the state of the input factors. For example, for a strong, helpful character, when the situation is considered serious, the character is most likely to run over and investigate, less likely to walk over and check out the situation, and least likely to ignore it. The corresponding utility values are accordingly set for this input combination, with the highest utility assigned to the run over action, a smaller one assigned to the walk over action and the lowest assigned to the ignore action, since the network evaluation will pick the action with the highest utility value. More details about how the utility values are assigned are given in Chapter 7.

As we mentioned previously, the seriousness node represents the character's interpretation of the emergency scene that it saw. This interpretation process is captured by a separate network (Figure 6.2). To select appropriate variables, we first consider the cues people look for when they try to judge whether a situation is serious. We then select those that can be observed in our characters, given their limited DI-Guy motor system. In our current implementation, we selected the age and gender of the patient. The agreeableness personality trait of the Big Five Model concerns how caring the character is about others.

This trait affects how characters will judge the seriousness of the situation. To model its effect, we create a chance node for each variable; namely, the seriousness, the age of the patient, the gender of the patient, and how caring the character is. The seriousness can take two states, *yes* or *no*, whereas the age of the patient can take three possible states, *young*, *prime age* or *old*. The gender of the patient can be either *male* or *female*, and the caring personality can be either *yes* or *no*. Since the assessment of the seriousness of the situation can explain observations being made about the age and gender of the patient as well as the character's caring personality, we add an arc from the seriousness node to the other three chance nodes. This completes the network structure. The conditional probabilities are set based on the seriousness judgement and what are the chances that the corresponding cues will be observed. That is, when the situation is deemed serious, how likely each possible state of the patient's age and gender will be, and how likely will the character be a caring one. For example, assuming the character generally regards the situation most serious when an old female patient is involved, the corresponding conditional probabilities need to be set higher than the ones for the other observation states. Following this intuition, $P(\text{Age} = \textit{old} \,|\, \text{Sr} = \textit{yes})$ is assigned the highest probability, while $P(\text{Age} = \textit{young} \,|\, \text{Sr} = \textit{yes})$ is set lower, and $P(\text{Age} = \textit{prime} \,|\, \text{Sr} = \textit{yes})$ is set the lowest. Similarly, $P(\text{Gender} = \textit{female} \,|\, \text{Sr} = \textit{yes})$ and $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{yes})$ are set to be higher than $P(\text{Gender} = \textit{male} \,|\, \text{Sr} = \textit{yes})$ and $P(\text{Care} = \textit{no} \,|\, \text{Sr} = \textit{yes})$, respectively.

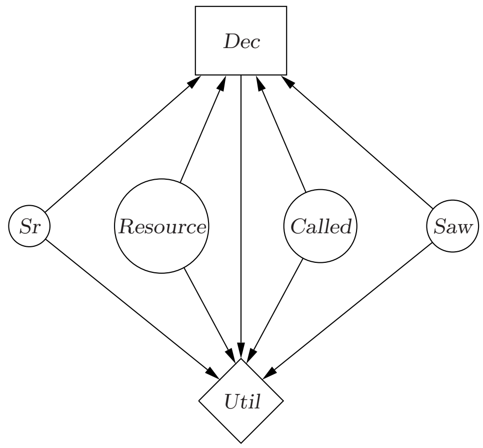
The aforementioned procedure constructs the networks needed to decide the initial reaction to an emergency situation. After the characters get close to the scene and investigated the situation, the more resourceful ones may decide to call for help from a law enforcement officer, or call on others to help. We approach this by considering what are the factors important to this decision making. The seriousness of the situation, how resourceful the character is, whether the character knows the whereabout of a law enforcement officer and whether the character believes help has already been called sprang to mind. So a chance node is set up for each of these four variables as depicted in

Figure 6.3. The possible actions in this case are not to call for help, or call others to help, or to fetch a police officer for help. A decision node needs to be added to represent this action selection, and a utility node is needed to specify the character's preferences. Arcs are added to link the four chance nodes to the decision and the utility node, and to link the decision node to the utility node similar to the network of Figure 6.1. The priors for the seriousness node comes from the evaluation of the network of Figure 6.2, whereas the priors for the variable representing if help has been fetched is again an interpretation result which can come from a separate network. The prior settings for the rest of the variables depend on the character's corresponding personality and internal factor, and the utility values are specified according to the designer's common sense (more explanations available in Chapter 7).

For a character to assess if someone else is fetching help, we need to think about the observable cues upon which one can base this judgment. Though our framework is general and can accommodate any available cues, we are limited by the possible motions that can be exhibited by the DI-Guy motor system. One indication of someone is fetching help is that they are running away from the scene. However, if the character is closer to the police or can run faster, then the character may still decide to go fetch help as it can reach help sooner. Therefore, whether another character, who is fetching help, is likely to summon help more quickly is also a relevant cue. Hence, we construct the network (Figure 6.4) by putting each of these three variables in chance nodes, and adding arcs from the Called node, which is to be interpreted, to the two nodes representing the cues to be observed. The corresponding conditional probabilities are set based on when help is called and the likelihood of observing the corresponding cues. The evaluation result obtained from this network is then inputted to the decision network in Figure 6.3.

Finally, for the characters who decide to observe the scene from close up, we must model when they will grow impatient and decide to leave. The factors to consider in reaching this decision include how patient the character is, whether the character is in

P(Sr)		P(Resource)		P(Called)		P(Saw)	
yes	no	yes	no	yes	no	yes	no
sr	1.0-sr	r	1.0-r	c	1.0-c	s	1.0-s

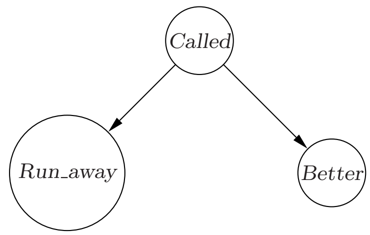


Sr=yes					Sr=no				
Res	Called	Saw	Dec	Util	Res	Called	Saw	Dec	Util
yes	yes	yes	no	5	yes	yes	yes	no	10
yes	yes	yes	ca	3	yes	yes	yes	ca	0
yes	yes	yes	cp	0	yes	yes	yes	cp	0
yes	yes	no	no	9	yes	yes	no	no	10
yes	yes	no	ca	7	yes	yes	no	ca	0
yes	yes	no	cp	0	yes	yes	no	cp	0
yes	no	yes	no	0	yes	no	yes	no	10
yes	no	yes	ca	5	yes	no	yes	ca	2
yes	no	yes	cp	10	yes	no	yes	cp	5
yes	no	no	no	0	yes	no	no	no	10
yes	no	no	ca	10	yes	no	no	ca	5
yes	no	no	cp	5	yes	no	no	cp	2
no	yes	yes	no	3	no	yes	yes	no	10
no	yes	yes	ca	2	no	yes	yes	ca	0
no	yes	yes	cp	2	no	yes	yes	cp	0
no	yes	no	no	7	no	yes	no	no	10
no	yes	no	ca	2	no	yes	no	ca	0
no	yes	no	cp	0	no	yes	no	cp	0
no	no	yes	no	3	no	no	yes	no	10
no	no	yes	ca	3	no	no	yes	ca	0
no	no	yes	cp	3	no	no	yes	cp	0
no	no	no	no	3	no	no	no	no	10
no	no	no	ca	2	no	no	no	ca	0
no	no	no	cp	0	no	no	no	cp	0

Figure 6.3: Network to decide whether to go fetch a police officer. In the CPTs, Sr: the seriousness of the situation, the parameter sr is the assessment made by the network shown in Figure 6.2, Resource: how resourceful the character is in finding help, Called: whether someone else is already seeking assistance, the parameter c is the assessment made by the network shown in Figure 6.4, Saw: if the character has noticed a law enforcement officer earlier, the parameter s is the character's corresponding state when this network is invoked. In the utility table, the character's decision Dec can assume the values of no (not calling for help), ca (call on passers-by to help), cp (go fetch a police officer).

haste, how long the character has been inactive at the scene with nothing they can do to help, whether someone else has already attempted to get help, and how much the character cares about others. Again, we build the network (Figure 6.5) by representing

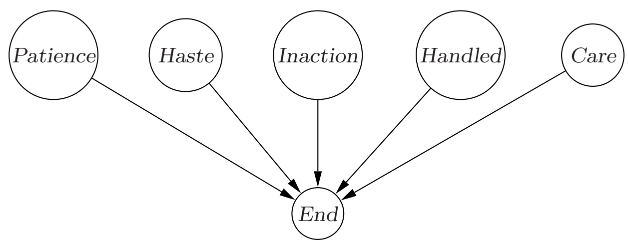
P(Called)	
yes	no
0.5	0.5



	P(R)		P(B)	
Called	yes	no	yes	no
yes	0.8	0.2	0.9	0.1
no	0.1	0.9	0.1	0.9

Figure 6.4: Network to assess if someone else is summoning the police. In the CPTs, R: whether someone else is running away from the scene, apparently to summon help, B: whether there are other characters who are also summoning help and are likely to summon help more quickly.

P(Patience)		P(Haste)		P(Inaction)		P(Handled)		P(Care)	
yes	no	yes	no	l	s	yes	no	yes	no
p	1.0-p	h	1.0-h	0.5	0.5	0.5	0.5	c	1.0-c



P=yes				P(End)		P=no				P(End)	
Haste	I	H	C	yes	no	Haste	I	H	C	yes	no
yes	l	yes	yes	0.7	0.3	yes	l	yes	yes	0.9	0.1
yes	l	yes	no	0.8	0.2	yes	l	yes	no	1.0	0.0
yes	l	no	yes	0.6	0.4	yes	l	no	yes	0.8	0.2
yes	l	no	no	0.7	0.3	yes	l	no	no	0.9	0.1
yes	s	yes	yes	0.5	0.5	yes	s	yes	yes	0.6	0.4
yes	s	yes	no	0.6	0.4	yes	s	yes	no	0.7	0.3
yes	s	no	yes	0.3	0.7	yes	s	no	yes	0.4	0.6
yes	s	no	no	0.4	0.6	yes	s	no	no	0.5	0.5
no	l	yes	yes	0.6	0.4	no	l	yes	yes	0.7	0.3
no	l	yes	no	0.7	0.3	no	l	yes	no	0.8	0.2
no	l	no	yes	0.51	0.49	no	l	no	yes	0.6	0.4
no	l	no	no	0.6	0.4	no	l	no	no	0.7	0.3
no	s	yes	yes	0.4	0.6	no	s	yes	yes	0.45	0.55
no	s	yes	no	0.4	0.6	no	s	yes	no	0.45	0.55
no	s	no	yes	0.0	1.0	no	s	no	yes	0.2	0.8
no	s	no	no	0.1	0.9	no	s	no	no	0.3	0.7

Figure 6.5: Network for deciding to end the emergency response behavior. In the CPTs, P: how patient the character is, Haste: if the character is in haste, I: how long has the character been present and unable to do much to help, assuming the values of long (l) or short (s), H: whether the situation has already been attended to, C: how much the character cares for others. The parameters p, h, c, are the character's corresponding personality traits and internal factors.

each variable with a chance node. In this case, since the decision is a simple yes/no decision whether to leave the scene, we can simplify the network by omitting the utility and decision node, and represent the decision with a chance node, specifying the preferences with the associated conditional probabilities. For example, since the character absolutely prefers to leave the scene when it is impatient, in a hurry, has not been able to do much to help for a long time and does not care for others that much, and when someone else has already fetched help, the corresponding conditional probability $P(\text{End} = \textit{yes} \,|\, \text{Patience} = \textit{no}, \text{Haste} = \textit{yes}, \text{Inaction} = \textit{long}, \text{Handled} = \textit{yes}, \text{Care} = \textit{no})$ is assigned a value of 1.0, while other conditional probabilities are set based on the designer's common sense. Chapter 7 provides additional details.

Thus, we have constructed all the networks our virtual characters use to decide how to respond to emergency situations. The probability and utility parameters in the networks are set based on the designer's intuitions, but they may need to be fine-tuned to truly reflect how the designer wants the characters to make decisions. In the next section, we will describe how this process can be completed.

6.1.2 Network Parameter Adjustments

The Netica software provides a GUI for defining and evaluating the networks. Hence, once the networks have been constructed, one can easily fine tune the parameter values through the GUI by trying various input combinations and observing the corresponding evaluation results in order to make the necessary adjustments.

Since our framework makes it easy to trace how a conclusion is drawn, when we see an undesirable result, we can quickly identify the cause and fix it. For example, consider the network of Figure 6.2. Suppose that we initially assigned $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{yes}) = 0.6$ and $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{no}) = 0.4$. As shown in Figure 6.6(a), trying the input combination of a male patient in his prime and a fully caring character, the situation is interpreted to be not serious. This is undesirable, since the character is caring and still considers the situation not to be serious, then for this kind of patient, no character will consider it serious. We want the fully caring character to consider the case serious even when the patient is a male in his prime. This evaluation result shows that we have not given enough influence to the caring variable. We need to increase the probability of observing this trait when the situation is considered serious, and similarly increase the probability of seeing no trace of this trait when the situation is considered not serious. Next, we try to assign $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{yes}) = 0.8$ and $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{no}) = 0.2$ (Figure 6.6(b)). We see that the probability for seriousness has increased for the same input combination, but not enough for the character to deem it serious, so we need to adjust the probabilities further. Finally, trying $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{yes}) = 0.9$ and $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{no}) = 0.1$ (Figure 6.6(c)), we obtain the desirable result. We can test this further on uncertain inputs. As illustrated in Figure 6.6(d) and 6.6(e), when there is some uncertainty in the caring variable (i.e., the character is somewhat caring, but not fully caring), the character may or may not regard the case involving a male patient in his prime as serious depending on the extent of the character's caring trait.

Our experience shows that a range of parameter values will work for the same decision process; that is, the probabilities do not have to be set precisely for the system to work well. To follow the same example, if we have set $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{yes}) = 0.95$ and $P(\text{Care} = \textit{yes} \,|\, \text{Sr} = \textit{no}) = 0.05$, we can still obtain desirable results. The difference is that with this setting, the character tends to think the situation as serious for a lower degree of caring trait. As shown in Figure 6.6(f), to obtain with this setting roughly the same seriousness assessment as in Figure 6.6(e), the caring quality needed is lower than with the previous setting, but they are both within a reasonable range. Therefore, the network works with a range of parameter settings, and our experience shows that fine tuning the parameters in this way is a quick and intuitive process. One can easily observe the result for each input combination and quickly identify the cause of problems if any, as it is easy to determine what contributes to the solution.

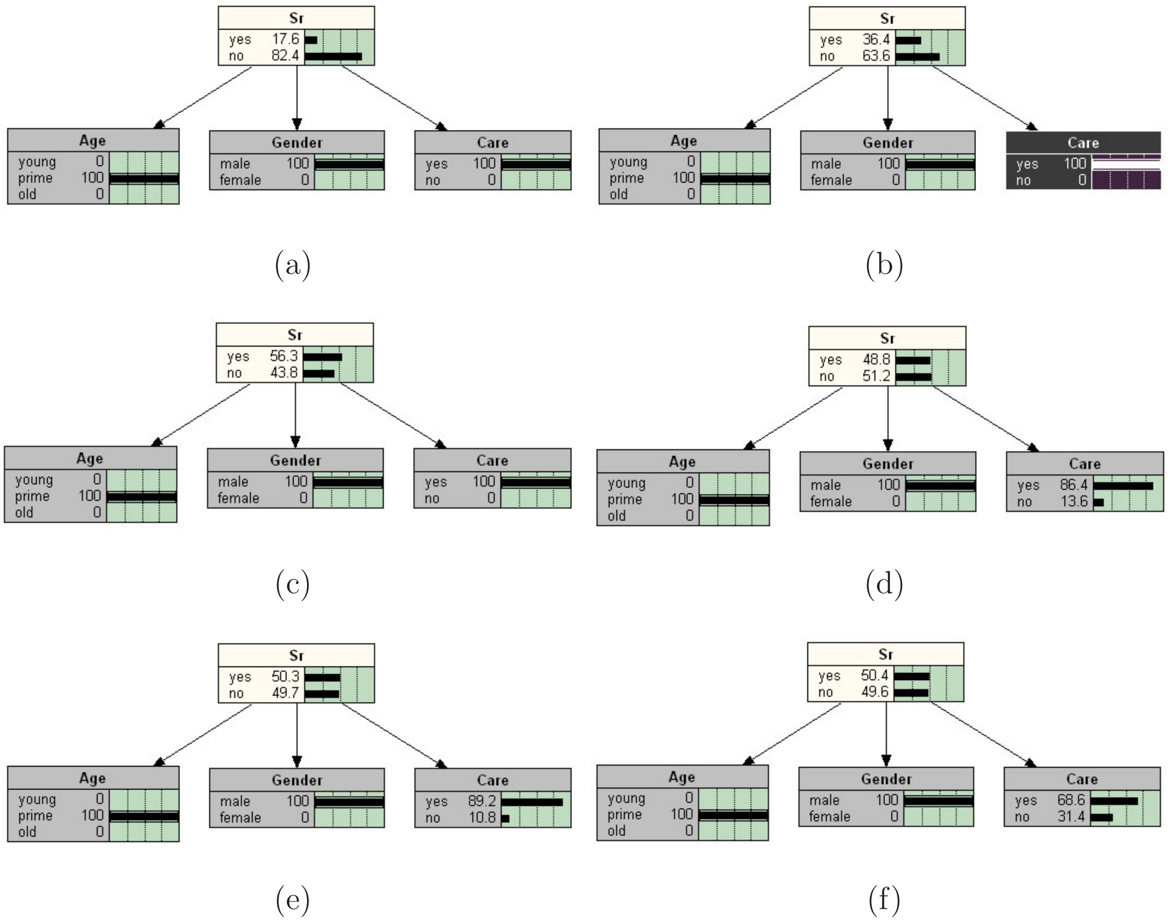


Figure 6.6: Parameter settings for the seriousness assessment network.

People think differently; even when making rational decisions, different people may have different standards for what is considered rational, and there are people who tend to act irrationally. Within our framework, the designer can easily shape their character's behaviors by changing the parameter settings. With the same network structure, by modifying various conditional probability and utility parameter settings, we can create a variety of characters that will decide to take quite different actions when faced with the same internal and external factors, even including "irrational characters" if that is what the designer wants. For example, if we want a strange character, who tends to ignore the situation when it is serious, and when the character really wants to help others and is courageous (i.e., the opposite of what a character with our current setting will choose to do), all we need to do is reverse the utility table entries to reflect this (i.e., give higher utility values to the ignore action and lower utility values to the run over and walk over actions, when Serious and Help nodes are in the *yes* state, and when Courage is in the *strong* state).

6.1.3 Network Flow

Given the defined networks, the decision process flows as follows: The character first uses the network shown in Figure 6.2 to assess the seriousness of the situation. Once this assessment is made, the character uses the decision network shown in Figure 6.1 to decide how to react. The more serious the character's assessment of the situation and the more altruistic the character, the more concerned the character will be about the incident, hence the stronger its reaction. The character's courage plays a role in determining if it can come close to the emergency scene. A timid character may choose not to get closer to the scene, and thus appear to ignore the incident.

The possible responses our model is capable of generating include: ignoring the incident, walking over to investigate, running over to investigate, and fetching help. Among the characters who have chosen to investigate, some may decide to seek assistance using the decision network shown in Figure 6.3. This decision depends on how serious the character regards the situation and how resourceful the character is in finding help, including whether or not the character has noticed a law enforcement officer earlier and can estimate the officer's whereabouts, and the character's assessment if someone else is already seeking assistance. This assessment is made by observing if someone else is running away from the emergency site and if it seem likely that the character can summon help more quickly (Figure 6.4).

Some characters may choose to leave the site of the incident after being present for some time and being unable to do much to help, a decision process implemented by the network of Figure 6.5. The contributing factors include how impatient the character is, if the character is in a hurry, the time that it has been idling at the scene, if someone is already summoning help, and how altruistic the character is.

6.1.4 Network Capabilities

Once the networks have been constructed as we have discussed in the previous sections, corresponding decisions based on various input combinations are automatically generated. Regardless of the number of influencing factors and how many possible states each factor can take, for any possible combination of them, the network is able to draw a corresponding conclusion. Moreover, our framework includes theoretically sound inference algorithms. They allow one to make interpretations based on evidence collected, including uncertain evidence.

Input factors include observations made about the character's environment and the character's own internal factors and personality traits. Differences in these factors lead to different decisions. For example, with the same networks, when the patient is a young female, a caring character believes that the situation is serious, and since the character is quite willing to help others and has strong courage, it decides to run over to investigate. When the patient is a male in his prime age, a non-caring character deems the situation as not serious and, being unwilling to help others, it chooses to ignore the case. There can be so many possible input factor combinations, but the same networks can automatically accommodate them. Table 6.1 shows the decisions generated for various input combinations obtained from the network of Figure 6.1.

In the network of Figure 6.1, all the input factors are binary. When we know for certain which state they are in, they are entered as hard evidence, and Table 6.1 illustrates the decisions made for such cases. However, most of the time, these factors are not strictly in one state or another. The assessment for seriousness that comes from the network of Figure 6.2 is represented with probability values to indicate different extents of seriousness. In the real world, people can vary in their willingness to help others and their level of courage, unlike a strictly binary classification. In our network, these assess-

Sr	Help	Courage	Decision
yes	yes	strong	run
yes	yes	weak	observe
yes	no	strong	observe
yes	no	weak	ignore
no	yes	strong	run
no	yes	weak	observe
no	no	strong	ignore
no	no	weak	ignore

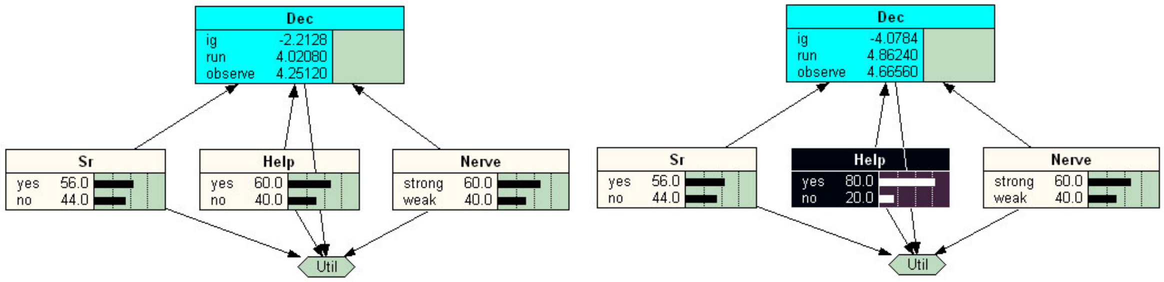
Table 6.1: Emergency response decisions based on various input combinations.

ments can be represented on a continuous scale from 0.0 to 1.0, and entered as the prior probability for their corresponding nodes. The network is capable of drawing inferences and making decisions incorporating these uncertainties. Hence, we can automatically obtain solutions with these uncertain data using our networks.

Figure 6.7 shows two examples where different decisions are made when the input factors take on different values. If we consider a value over 50% as being affirmative, in both cases illustrated here the situation is judged to be serious, the character is willing to help others, and has high courage. The only difference is the extent of wanting to help others, the one to the right having a higher degree, yet they end up choosing to act differently. The character to the left decides to walk over to investigate, while the more helpful character to the right decides to run over. The richness in various inputs that can be incorporated and the ability to handle uncertainties comes naturally with the networks constructed.

6.1.5 Network Extension

Since our framework offers a modular representation, it is easy to make extensions. When we need to add a factor that is to be considered in a decision process, we must add a node for it in the corresponding network. For our applications, the addition of a variable can be regarded as the consideration of additional causal relationships associated with this new variable. By connecting this new variable with arcs corresponding to these causal



(A) (B)

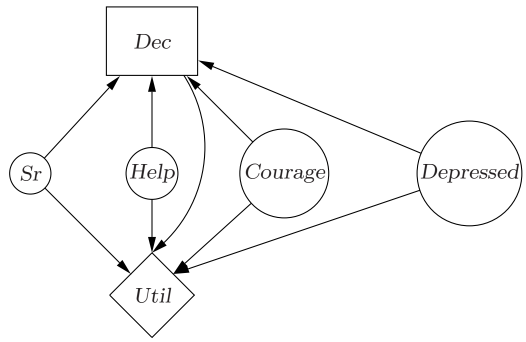
Figure 6.7: Emergency response decision with different degrees of input factors.

relations, the network structure will not be violated. This extension process is a natural mapping of the designer's intuition and commonsense knowledge. If a variable that is added to the network is not connected to any utility node, there is no need to modify the utility values in the network. Even if it is connected to a utility node, it may not be necessary to modify all existing utility values, as we will demonstrate in an example below. If the added variable is a leaf node, only conditional probabilities for it given its parents need be specified. If it is a node parenting to others, then in addition to its own conditional probabilities, the conditional probabilities for the children nodes of this new node must be adjusted to reflect the change. In both cases, only variables with links to the new variable are affected, which is typically a small part of the entire network.

For example, suppose we want to consider the effect of emotion on the emergency response behavior. A depressed character that is feeling quite low may lose interest in other things, including helping others, when normally the character would. This involves adding to the network of Figure 6.1 a new chance node, Depress, to model the emotional depression. This node can take a value of *yes* or *no*. As shown in Figure 6.8, links to the utility and decision nodes are added for Depress, since it is also an input factor for the decision process. When Depress is added in the Netica software, Netica automatically updates the utility table with this additional variable and duplicates the original entries to accommodate the additional entries brought by the new variable. That is, for both values of Depress, the corresponding utility values are copied from the original entry when Depress was absent. To complete the extension, the designer just needs to adjust these utility values to reflect the effect of the added emotional component. Assuming that when the character is not depressed, it should act the same as before we included this emotional factor, hence the utility values when Depress = *No* should remain the same, we only need to update the utility values when Depress = *Yes*. As shown in the utility table of Figure 6.8, we assume that when the character is depressed, it tends to respond less actively; hence, there is a generally higher chance of ignoring the scene rather than reacting more responsively such as running over to investigate. The belief-bar diagram in Figure 6.9 illustrates a case where under the same conditions for seriousness assessment, how helpful the character is and how strong the character's courage is, without considering the emotional depression, the character decides to walk over to the scene, whereas with the added emotional factor, a depressed character chooses to ignore the incident. If the designer wants a character, who would become more sympathetic towards others when feeling depressed, and hence become more inclined to respond actively regarding an emergency situation, one need only modify the corresponding utility values to reflect this; i.e., when Depressed is in the *yes* state, give higher utility values to the *run* and *obs* actions and lower utility value to the *ig* action.

Suppose that we want to model multiple emotional factors. We can reduce the number of entries need to be added to the utility table by building a subnetwork, where an overall effect from the emotional components is assessed first, and only add that effect to the utility table. For example, if we want to consider anger and happiness in addition to depression, we can set up a chance node corresponding to each emotion, and introduce an additional chance node to represent the overall effect these emotional factors will have on the final decision, which can take on the state of Active (prompting the character to be more active in responding to the emergency situation) and Inactive (prompting the character to be less active in responding). This overall effect node is then linked to the

P(Sr)		P(Help)		P(Courage)		P(Depressed)	
yes	no	yes	no	s	w	yes	no
sr	1.0-sr	he	1.0-he	n	1.0-n	d	1.0-d

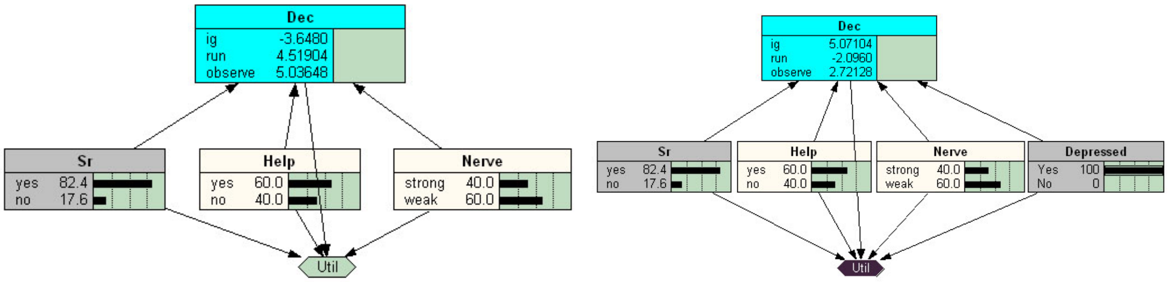


Depressed = No (**these entries are copied from the original table before extension**)								Depressed = Yes (**these entries are added for extension**)							
Sr=yes				Sr=no				Sr=yes				Sr=no			
Help	Courage	Dec	Util	Help	Courage	Dec	Util	Help	Courage	Dec	Util	Help	Courage	Dec	Util
yes	s	ig	-10	yes	s	ig	-5	yes	s	ig	0	yes	s	ig	7
yes	s	run	8	yes	s	run	5	yes	s	run	5	yes	s	run	-5
yes	s	obs	6	yes	s	obs	3	yes	s	obs	6	yes	s	obs	5
yes	w	ig	-8	yes	w	ig	3	yes	w	ig	3	yes	w	ig	8
yes	w	run	6	yes	w	run	2	yes	w	run	0	yes	w	run	-6
yes	w	obs	7	yes	w	obs	4	yes	w	obs	4	yes	w	obs	2
no	s	ig	1	no	s	ig	5	no	s	ig	7	no	s	ig	9
no	s	run	3	no	s	run	1	no	s	run	-5	no	s	run	-8
no	s	obs	5	no	s	obs	2	no	s	obs	2	no	s	obs	-3
no	w	ig	3	no	w	ig	6	no	w	ig	9	no	w	ig	10
no	w	run	1	no	w	run	0	no	w	run	-6	no	w	run	-10
no	w	obs	2	no	w	obs	2	no	w	obs	0	no	w	obs	-4

Figure 6.8: Extension of the network of Figure 6.1 to include modeling of the depressed emotion. The necessary work involves adding the node Depressed, and its associated links. The corresponding addition to the conditional probability table and the utility table are listed to the right of the triple column separator, whereas the original entries are listed to the left. The parameter d is an indication of how depressed the character is, it is scaled from 0 to 1.

utility and the decision node, in place of the node representing the depressed emotion, as shown in Figure 6.10.

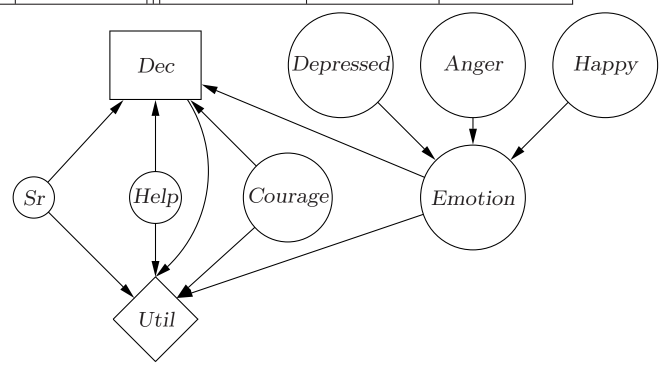
To ensure the compact size of individual networks in order to facilitate extension and modification and to keep the computation cost at bay, we arrange the networks hierarchically. Any self-contained network structure, such as the assessment of the seriousness of an emergency situation, can be implemented as a separate network and its result reused in any other network involving this factor. In our experience, the size of every network that we have needed to implement is small, so the specification for the conditional probability distributions and the utility values was by no means time consuming. Extension can be easily accomplished as the changes needed are isolated to the concerned network(s), with



(A) (B)

Figure 6.9: Belief-bar diagram to illustrate the effect of the Depressed emotion.

P(Sr)		P(Help)		P(Courage)		P(Depressed)		P(Anger)		P(Happy)	
yes	no	yes	no	s	w	yes	no	yes	no	yes	no
sr	1.0-sr	he	1.0-he	n	1.0-n	d	1.0-d	a	1.0-a	h	1.0-h



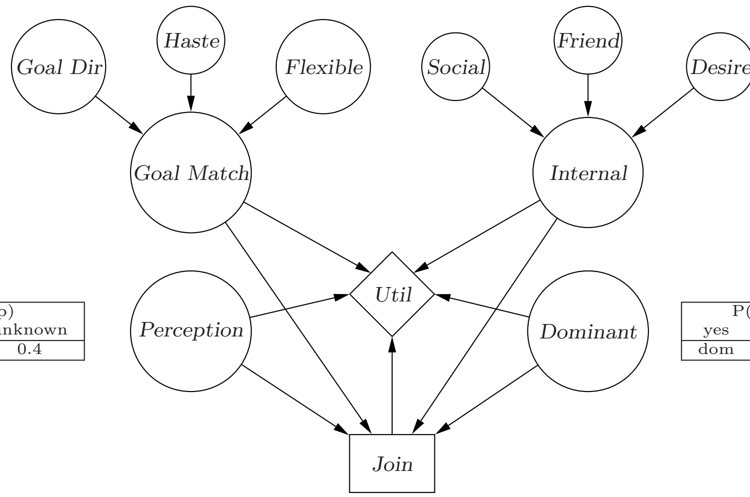
Depressed=yes		P(Emotion)		Depressed=no		P(Emotion)	
Anger	Happy	Inactive	Active	Anger	Happy	Inactive	Active
yes	yes			yes	yes		
yes	no			yes	no		
no	yes			no	yes		
no	no			no	no		

Emotion = Inactive								Emotion = Active							
Help	Courage	Dec	Util	Help	Courage	Dec	Util	Help	Courage	Dec	Util	Help	Courage	Dec	Util
yes	s	ig		yes	s	ig		yes	s	ig		yes	s	ig	
yes	s	run		yes	s	run		yes	s	run		yes	s	run	
yes	s	obs		yes	s	obs		yes	s	obs		yes	s	obs	
yes	w	ig		yes	w	ig		yes	w	ig		yes	w	ig	
yes	w	run		yes	w	run		yes	w	run		yes	w	run	
yes	w	obs		yes	w	obs		yes	w	obs		yes	w	obs	
no	s	ig		no	s	ig		no	s	ig		no	s	ig	
no	s	run		no	s	run		no	s	run		no	s	run	
no	s	obs		no	s	obs		no	s	obs		no	s	obs	
no	w	ig		no	w	ig		no	w	ig		no	w	ig	
no	w	run		no	w	run		no	w	run		no	w	run	
no	w	obs		no	w	obs		no	w	obs		no	w	obs	

Figure 6.10: Extension of the network of Figure 6.1 to include modeling of the depression, anger, and happiness emotions. The necessary work involves adding the three nodes and a Emotion node to assess the overall effect of the emotional components combined, their associated links and conditional probabilities, and the update to the utility table entries. The parameter d is an indication of how depressed the character is, a is how angry the character is, and h is how happy the character is. The unfilled table entries indicate the ones need to be filled or adjusted to complete the extension.

P(GD)		P(H)		P(Fl)	
yes	no	yes	no	yes	no
0.5	0.5	h	1.0-h	fl	1.0-fl

P(S)		P(F)		P(D)	
yes	no	yes	no	yes	no
s	1.0-s	f	1.0-f	d[partner]	1.0-d[partner]



P(Percep)		
yes	no	unknown
0.3	0.3	0.4

P(Dom)	
yes	no
dom	1.0-dom

GD=yes		P(GM)		GD=no		P(GM)	
H	Fl	yes	no	H	Fl	yes	no
yes	yes	0.5	0.5	yes	yes	0.5	0.5
yes	no	0.2	0.8	yes	no	0.0	1.0
no	yes	1.0	0.0	no	yes	0.9	0.1
no	no	0.8	0.2	no	no	0.1	0.9

S=yes		P(Internal)		S=no		P(Internal)	
F	D	yes	no	F	D	yes	no
yes	yes	1.0	0.0	yes	yes	0.9	0.1
yes	no	0.2	0.8	yes	no	0.1	0.9
no	yes	0.3	0.7	no	yes	0.2	0.8
no	no	0.05	0.95	no	no	0.0	1.0

Dom=yes					Dom=no				
Per	GM	Int	Join	Util	Per	GM	Int	Join	Util
yes	yes	yes	yes	10	yes	yes	yes	yes	8
yes	yes	yes	no	-10	yes	yes	yes	no	-8
yes	yes	no	yes	-1	yes	yes	no	yes	2
yes	yes	no	no	5	yes	yes	no	no	6
yes	no	yes	yes	-3	yes	no	yes	yes	-1
yes	no	yes	no	5	yes	no	yes	no	3
yes	no	no	yes	-9	yes	no	no	yes	-4
yes	no	no	no	9	yes	no	no	no	4
no	yes	yes	yes	6	no	yes	yes	yes	1
no	yes	yes	no	3	no	yes	yes	no	-1
no	yes	no	yes	-10	no	yes	no	yes	-6
no	yes	no	no	9	no	yes	no	no	6
no	no	yes	yes	1	no	no	yes	yes	-5
no	no	yes	no	6	no	no	yes	no	5
no	no	no	yes	-10	no	no	no	yes	-10
no	no	no	no	10	no	no	no	no	10
uk	yes	yes	yes	9	uk	yes	yes	yes	8
uk	yes	yes	no	-7	uk	yes	yes	no	-6
uk	yes	no	yes	-5	uk	yes	no	yes	-2
uk	yes	no	no	6	uk	yes	no	no	5
uk	no	yes	yes	2	uk	no	yes	yes	1
uk	no	yes	no	4	uk	no	yes	no	3
uk	no	no	yes	-7	uk	no	no	yes	-4
uk	no	no	no	7	uk	no	no	no	4

Figure 6.11: Network to decide whether to form a partnership. In the CPTs, GD: if character A's goal direction and character B's walking direction is within certain range, H: if A is in haste, Fl: if A is flexible in changing its current goal, S: if A is social, F: A's friendliness with B, D: A's desire to partner with another character, Percep: A's perception of whether B wants to partner with A, which can take on the values of yes, no, or unknown, Dom: if A is a dominant character, GM: if A feels its goal matches that of B, Internal: if A is intended to partner with B. The parameters h, fl, s, d[partner] and dom are A's corresponding internal factors and personality traits.

the addition of the corresponding nodes and links. We believe that it is a much more systematic and quicker process than trying to obtain something similar with rules.

6.2 Partnering Behavior

It is natural for friends meeting unexpectedly to form partnerships if circumstances permit. We have developed a partnering model that simulates such behavior.

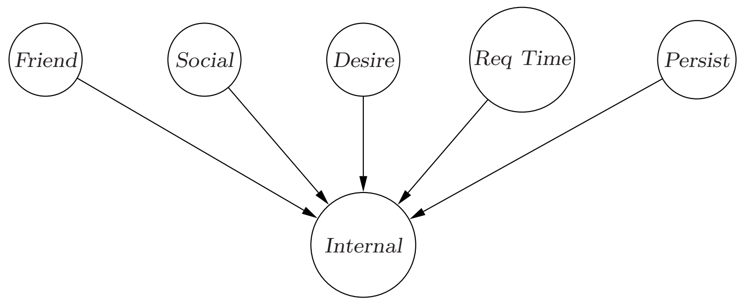
There are several possible scenarios: when a character sees a friend character ahead, it needs to decide whether to catch up and try to partner with the other character. Then, the other character has to decide whether to accept the advance. When two characters happen to run into each other, they may have a chat and then decide if they want to partner. In the first case, the initiating character A must first decide if it wants to approach and try to form a partnership with the friend character B. This decision must take into consideration A's intention, personality, if A feels its goal matches that of B, and its perception of B's intention. Figure 6.11 shows the decision network for deciding whether or not to form a partnership with the potential partner.

Once A decides to form a partnership with B, it will try to approach and make its partnership advance. Then B must decide whether to accept the advance. As shown in Figure 6.12, how friendly B feels about A, if B is social, and B's desire, all affect the decision. In addition, how persistent B is in pursuing its own intention and how long A has persisted in making the advance also influence B's reaction.

As B decides, A observes B and assesses if B has accepted its advance. Figure 6.13 shows the network that makes this interpretation. Not all the observed data have equal reliability. For example, when trying to interpret B's response to an advance, B turning to look at A and stopping are much stronger cues than B simply glancing at A. When B accepts A's request, B may or may not choose to face A directly. To accommodate this difference in reliability of the observed data, adjustments are made in the network, hence the presence of the additional reliability chance node.

If B accepts A's request, the two characters will form a partnership and proceed. Should B refuse the request, A must then decide how it will deal with the rejection—give up or continue its persuasion effort (Figure 6.14). This cycle of A making an advance,

P(Friend)		P(Social)		P(Desire)		P(Req_time)		P(Persist)	
yes	no	yes	no	yes	no	l	s	yes	no
f	1.0-f	s	1.0-s	desire[partner]	1.0-desire[partner]	0.5	0.5	per	1.0-per



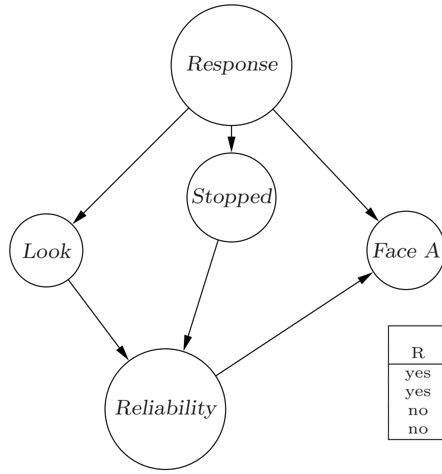
Friend=yes				P(Internal)		Friend=no				P(Internal)	
S	D	RT	P	yes	no	S	D	RT	P	yes	no
yes	yes	l	yes	0.9	0.1	yes	yes	l	yes	0.2	0.8
yes	yes	l	no	1.0	0.0	yes	yes	l	no	0.3	0.7
yes	yes	s	yes	0.8	0.2	yes	yes	s	yes	0.1	0.9
yes	yes	s	no	0.9	0.1	yes	yes	s	no	0.1	0.9
yes	no	l	yes	0.6	0.4	yes	no	l	yes	0.0	1.0
yes	no	l	no	0.7	0.3	yes	no	l	no	0.1	0.9
yes	no	s	yes	0.3	0.7	yes	no	s	yes	0.0	1.0
yes	no	s	no	0.4	0.6	yes	no	s	no	0.0	1.0
no	yes	l	yes	0.7	0.3	no	yes	l	yes	0.1	0.9
no	yes	l	no	0.7	0.3	no	yes	l	no	0.2	0.8
no	yes	s	yes	0.7	0.3	no	yes	s	yes	0.0	1.0
no	yes	s	no	0.7	0.3	no	yes	s	no	0.1	0.9
no	no	l	yes	0.1	0.9	no	no	l	yes	0.0	1.0
no	no	l	no	0.2	0.8	no	no	l	no	0.1	0.9
no	no	s	yes	0.1	0.9	no	no	s	yes	0.0	1.0
no	no	s	no	0.1	0.9	no	no	s	no	0.0	1.0

Figure 6.12: Network to assess whether to accept a partnering request. In the CPTs, S: how social the character is, D: B's desire to form a partnership, RT: how long the other character has persisted in making the partnership advance, which can take on the values of long (l) or short (s), P: how persistent the character is in pursuing its own intention. The parameters f, s, desire[partner], and per are the character's corresponding internal factors and personality traits.

B deciding whether to accept it, A's interpretation of B's response, and decision about what to do next continues until either B is persuaded to accept the partnership offer or A gives up.

Personality and other internal factors, such as haste, play an important role in the decisions we make as humans. The decision network framework provides a convenient formalism for incorporating such factors and enabling them to influence behavior. Consider the partnership decision, for example. To infer whether the character feels that its goal matches that of its potential partner, beyond their immediate goals, internal factors such as whether or not the character is in a hurry and how flexible the character is in changing its current goal to accommodate its potential partner's also play a role. The

P(R)	
yes	no
0.5	0.5



R	P(LA) yes	P(LA) no
yes	0.8	0.2
no	0.2	0.8

R	P(ST) yes	P(ST) no
yes	0.8	0.2
no	0.2	0.8

R	Rel	P(FA) yes	P(FA) no
yes	yes	0.8	0.2
yes	no	0.55	0.45
no	yes	0.2	0.8
no	no	0.45	0.55

LA=yes ST	P(Reliability) yes	P(Reliability) no	LA=no ST	P(Reliability) yes	P(Reliability) no
yes	1.0	0.0	yes	0.0	1.0
no	0.0	1.0	no	0.0	1.0

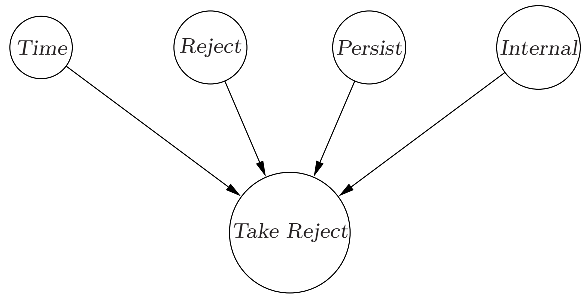
Figure 6.13: Network to interpret potential partner's response to A's partnering request. In the CPTs, R: interpreted response, LA: if B is looking at A, ST: if B has stopped, FA: if B is facing A, Rel: the reliability factor.

character's own intention of whether to form a partnership depends on how social the character is, how much desire it has in joining a partnership and how friendly it feels about the potential partner. If the character has a dominant personality, it tends to weigh its own intention and desire more heavily when making a decision. So, it is less likely for a dominant character than a submissive one to form a partnership when it feels its goal does not align with that of its potential partner. Similarly, how persistent a character is and how well it takes rejection affect the character's reaction to the potential partner's refusal. If the character is quite stubborn and can take rejection well, it will insist on making partnership requests; otherwise, it will give up more easily.

6.3 Acquaintance Behavior

Interpretation becomes more important in interactions between pedestrians. For example, when two acquaintances meet on the street, how they react to one another depends not only on their own intentions, but also on their interpretation of what the other will

P(TT)		P(Reject)		P(Persist)		P(Internal)	
yes	no	yes	no	yes	no	yes	no
0.5	0.5	r	1.0-r	p	1.0-p	0.5	0.5



TT=yes			P(TR)		TT=no			P(TR)	
R	P	Int	yes	no	R	P	Int	yes	no
yes	yes	yes	0.0	1.0	yes	yes	yes	1.0	0.0
yes	yes	no	0.0	1.0	yes	yes	no	0.3	0.7
yes	no	yes	0.0	1.0	yes	no	yes	0.6	0.4
yes	no	no	0.0	1.0	yes	no	no	0.2	0.8
no	yes	yes	0.0	1.0	no	yes	yes	0.5	0.5
no	yes	no	0.0	1.0	no	yes	no	0.3	0.7
no	no	yes	0.0	1.0	no	no	yes	0.5	0.5
no	no	no	0.0	1.0	no	no	no	0.1	0.9

Figure 6.14: Network to assess how to take rejection. In the CPTs, TT: if the time that the character has been making partnership advance has exceeded certain time threshold, R: if the character can take rejection well, P: if the character is persistent in pursuing its current goal, Int: if the character's intention in forming a partnership with B is strong, TR: how the character should deal with the rejection, yes means the character will continue its persuasion effort, no means the character will give up. The parameters r and p are the character's corresponding personality traits.

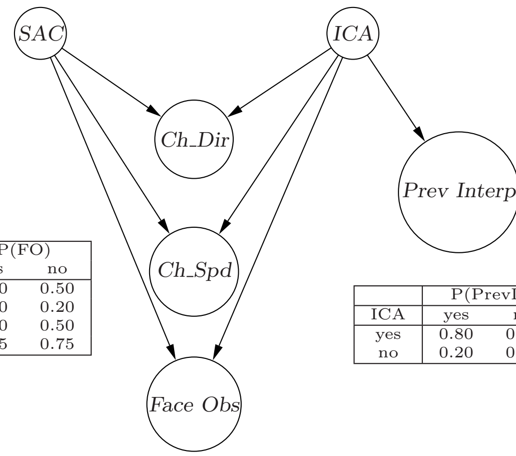
do. We have implemented an acquaintance behavior model to simulate this interaction.

When two acquainted characters meet, they will choose among talking to each other, acknowledging one another without stopping to chat, or ignoring one another. For the acquaintance model, two pedestrians cannot talk with each other unless both are willing to talk; hence, a reasonably accurate interpretation is needed. Given the current limitations of DI-Guy, for now the characters are mainly observing each other's walking direction, speed change, and gaze direction to make the judgment.

Character A's interpretation of character B's intention is divided into an assessment of whether or not B is just starting to engage with character A (showing an intention to talk, or to greet), or was already in the process of engaging with A, as they exhibit different cues. When B is just starting to engage with A, then B would be looking at A; if B wants to talk with A, then B would most likely start to walk towards A and slow

P(SAC=yes)	P(SAC=no)
0.50	0.50

P(ICA=yes)	P(ICA=no)
0.50	0.50



		P(CD)		P(CS)		P(FO)	
SAC	ICA	yes	no	yes	no	yes	no
yes	yes	0.50	0.50	0.50	0.50	0.50	0.50
yes	no	0.80	0.20	0.80	0.20	0.80	0.20
no	yes	0.20	0.80	0.20	0.80	0.50	0.50
no	no	0.50	0.50	0.50	0.50	0.25	0.75

	P(PrevIn	
ICA	yes	no
yes	0.80	0.20
no	0.20	0.80

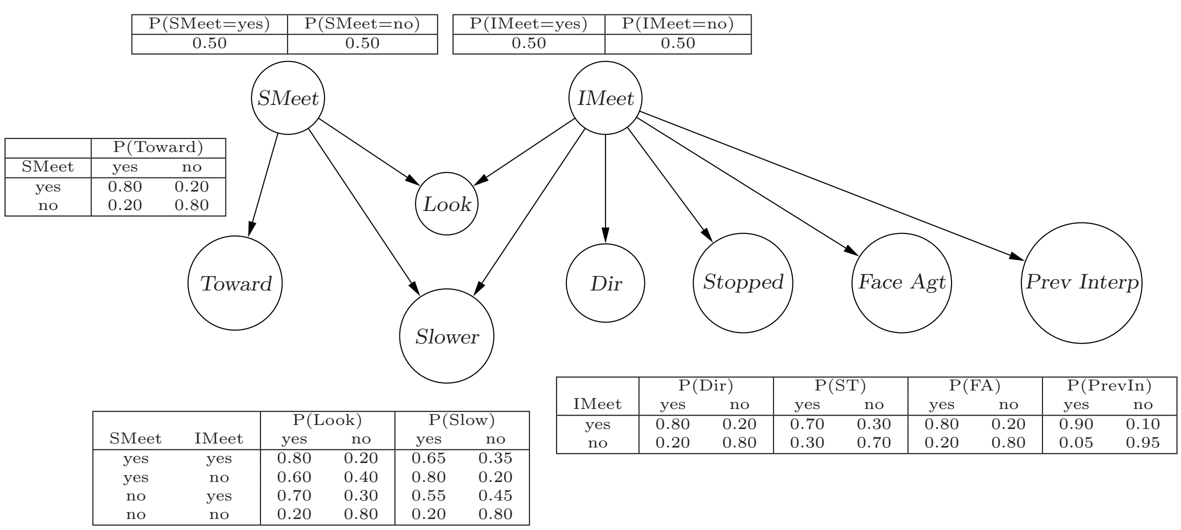
Figure 6.15: Network to assess if the other character is starting to avoid a collision or is already in collision avoidance. The meanings of the symbols in the CPTs are as follows: SAC: the interpretation that the other character is starting to avoid a collision, ICA: is already in collision avoidance, CD: changing direction, CS: changing speed, FO: facing an obstacle. Prev Interp denotes the interpretation the character made previously about the other character regarding if it is already in collision avoidance.

down. The absence of these cues is evidence that B intends to ignore A. When B is in the process of carrying out its intended action with A (for example, trying to talk with A) B may already be walking towards A, hence does not change direction at the moment. The prerequisite is that the previous interpretation already indicated that B intended to engage with A.

The difficulty in making such an interpretation is further complicated by the fact that B may intend to engage with A, but is temporarily interrupted by the need to avoid collisions with third parties. Hence, an assessment is required first of whether or not B is in collision avoidance mode. Such an assessment is made based on change in direction, change in speed, whether B is facing obstacles, and determination if B is just starting to avoid collision or is engaged in collision avoidance. B talking with a third party will also prevent B from talking with A. Figs. 6.15, 6.16, and 6.17 show the networks for interpreting B's current action, whether it be meeting with others, avoiding collision, or just walking.

Once this interpretation is made, further decisions are made based on it. If B is talking with a third party, no further action is taken (in future work, it should be straightforward to implement the possibility of A joining the conversation). If B is engaged in avoiding a collision, the prior interpretation for B is maintained. In the absence of any special action detection, A will continue to assess B's intention (Figure 6.18). In addition, A evaluates its own intended action based on its friendliness with B, its haste, and how much it desires to greet or talk to B.

Figure 6.19 depicts the decision network that determines the action to be taken. The distance factor is added since the character may be uncertain about the interpretation and, hence, may take several time steps to decide. In the meantime, it is not committing to any action. However, as the two characters approach each other to within a certain proximity, a decision must be made based on the information available at the time. The distance factor indicates if this proximity threshold has been met. The utility function



P(SMeet=yes)	P(SMeet=no)
0.50	0.50

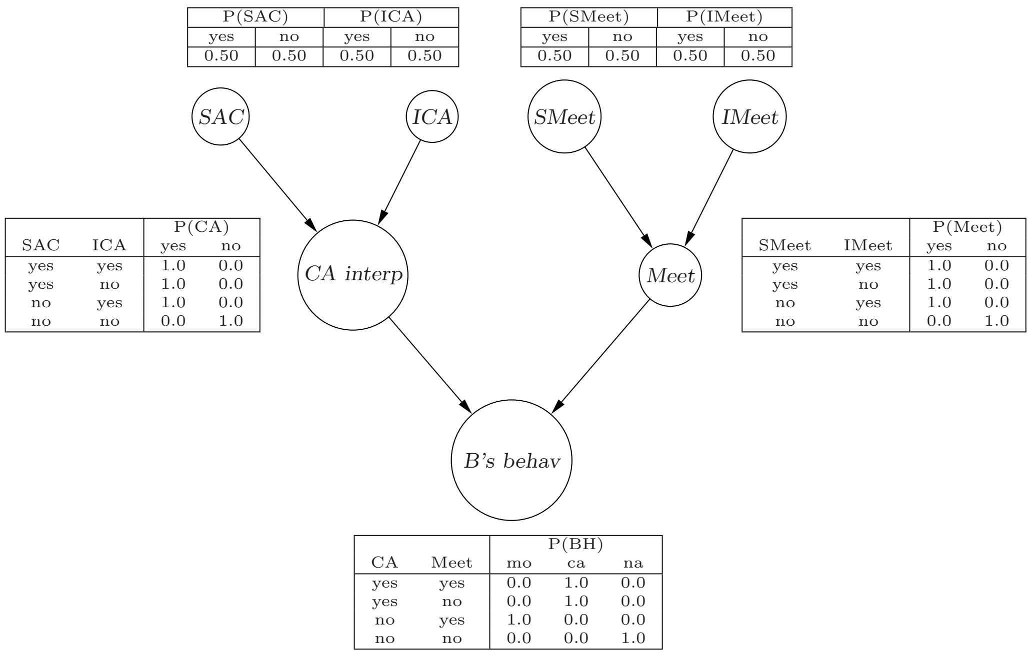
P(IMeet=yes)	P(IMeet=no)
0.50	0.50

	P(Toward)	
SMeet	yes	no
yes	0.80	0.20
no	0.20	0.80

		P(Look)		P(Slow)	
SMeet	IMeet	yes	no	yes	no
yes	yes	0.80	0.20	0.65	0.35
yes	no	0.60	0.40	0.80	0.20
no	yes	0.70	0.30	0.55	0.45
no	no	0.20	0.80	0.20	0.80

	P(Dir)		P(ST)		P(FA)		P(PrevIn)	
IMeet	yes	no	yes	no	yes	no	yes	no
yes	0.80	0.20	0.70	0.30	0.80	0.20	0.90	0.10
no	0.20	0.80	0.30	0.70	0.20	0.80	0.05	0.95

Figure 6.16: Network to assess if the other character is starting to meet another character or is already meeting another character. In the CPTs, SMeet: the interpretation that the other character is starting to meet a third party, IMeet: is already meeting a third party, ST: is stopped, and FA: is facing another character in a certain proximity. Prev Interp denotes the interpretation the character made previously about the other character regarding if it is meeting another character.



P(SAC)		P(ICA)	
yes	no	yes	no
0.50	0.50	0.50	0.50

P(SMeet)		P(IMeet)	
yes	no	yes	no
0.50	0.50	0.50	0.50

		P(CA)	
SAC	ICA	yes	no
yes	yes	1.0	0.0
yes	no	1.0	0.0
no	yes	1.0	0.0
no	no	0.0	1.0

		P(Meet)	
SMeet	IMeet	yes	no
yes	yes	1.0	0.0
yes	no	1.0	0.0
no	yes	1.0	0.0
no	no	0.0	1.0

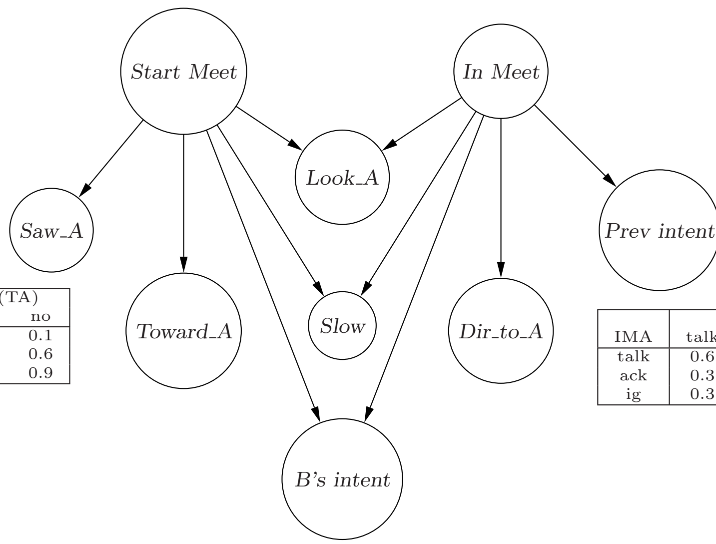
		P(BH)		
CA	Meet	mo	ca	na
yes	yes	0.0	1.0	0.0
yes	no	0.0	1.0	0.0
no	yes	1.0	0.0	0.0
no	no	0.0	0.0	1.0

Figure 6.17: Network to interpret the behavior in which the other character is engaged. In the CPTs, CA: the interpretation of whether the other character B is in collision avoidance, Meet: the interpretation of whether B is meeting other characters, and BH: the behavior that B is engaged in. (mo: meeting other characters, ca: in collision avoidance, na: not in any special action).

balances the internal desire with the interpreted intention of the other party. The decision can be to talk, greet, or ignore the other party, or uncertain, which means that further observation is necessary.

P(SMA)		
talk	ack	ig
0.30	0.30	0.40

P(IMA)		
talk	ack	ig
0.30	0.30	0.40



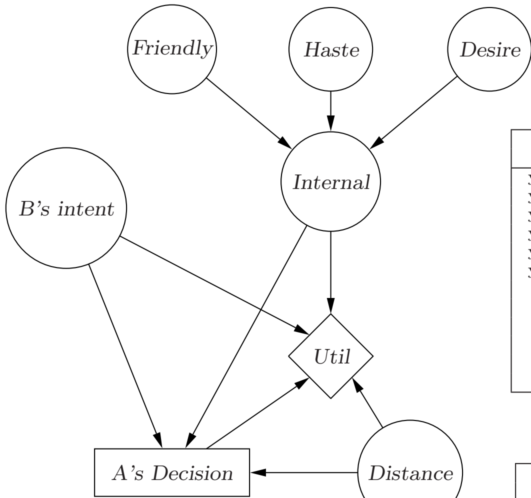
	P(SA)		P(TA)	
SMA	yes	no	yes	no
talk	0.9	0.1	0.9	0.1
ack	0.9	0.1	0.4	0.6
ig	0.5	0.5	0.1	0.9

	P(Prev)			P(DTA)	
IMA	talk	ack	ig	yes	no
talk	0.6	0.3	0.1	0.9	0.1
ack	0.3	0.5	0.2	0.4	0.6
ig	0.3	0.3	0.4	0.1	0.9

		P(LA)		P(SD)		P(BI)		
SMA	IMA	yes	no	yes	no	talk	ack	ig
talk	talk	0.9	0.1	0.9	0.1	1.0	0.0	0.0
talk	ack	0.7	0.3	0.7	0.3	0.7	0.3	0.0
talk	ig	0.3	0.7	0.5	0.5	0.6	0.3	0.1
ack	talk	0.7	0.3	0.6	0.4	0.5	0.4	0.1
ack	ack	0.6	0.4	0.5	0.5	0.0	1.0	0.0
ack	ig	0.5	0.5	0.3	0.7	0.2	0.5	0.3
ig	talk	0.55	0.45	0.6	0.4	0.6	0.2	0.2
ig	ack	0.5	0.5	0.3	0.7	0.1	0.5	0.4
ig	ig	0.1	0.9	0.1	0.9	0.0	0.0	1.0

Figure 6.18: Network to interpret the other character's intended action. In the CPTs, SMA : B is starting to engage with character A, which can take on the values of talk (stopping to chat), ack (acknowledging) or ig (ignoring), SA : B has seen character A, TA : B is turning towards A, LA : B is looking at A, SD : B is slowing down, BI : B's intent, IMA: B is already engaged with A, Prev : previous interpretation of B's intent, DTA : B is walking towards A.

P(Friendly)		P(Haste)		P(Desire)		
yes	no	yes	no	talk	ack	ig
f	1.0-f	h	1.0-h	desire[talk]	desire[ack]	1.0-desire[talk+ack]



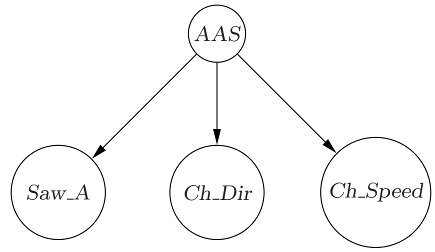
F	H	D	P(Int) talk	ack	ig
yes	yes	talk	0.5	0.4	0.1
yes	yes	ack	0.1	0.8	0.1
yes	yes	ig	0.1	0.6	0.3
yes	no	talk	0.8	0.1	0.1
yes	no	ack	0.3	0.6	0.1
yes	no	ig	0.1	0.4	0.5
no	yes	talk	0.2	0.4	0.4
no	yes	ack	0.1	0.5	0.4
no	yes	ig	0.1	0.1	0.8
no	no	talk	0.3	0.4	0.3
no	no	ack	0.2	0.5	0.3
no	no	ig	0.1	0.2	0.7

P(Dis)	
yes	no
0.5	0.5

Dis=yes				Dis=no			
BI	Int	Dec	Util	BI	Int	Dec	Util
talk	talk	talk	ut+10	talk	talk	talk	ut+10
talk	talk	ack	ua+5	talk	talk	ack	ua+5
talk	ack	talk	ut+7	talk	talk	obs	7
talk	ack	ack	ua+10	talk	ack	talk	ut+7
talk	ig	talk	ut	talk	ack	ack	ua+9
talk	ig	ack	ua+3	talk	ack	obs	10
talk	ig	ig	5	talk	ig	talk	ut
ack	talk	talk	ut+3	talk	ig	ack	ua+2
ack	talk	ack	ua+6	talk	ig	ig	5
ack	ack	talk	ut	talk	ig	obs	8
ack	ack	ack	ua+10	ack	talk	talk	ut+5
ack	ig	talk	ut	ack	talk	ack	ua+5
ack	ig	ack	ua	ack	talk	obs	10
ack	ig	ig	6	ack	ack	talk	ut
ig	talk	talk	ut	ack	ack	ack	ua+10
ig	talk	ack	ua+8	ack	ack	obs	6
ig	talk	ig	3	ack	ig	talk	ut
ig	ack	talk	ut	ack	ig	ack	ua
ig	ack	ack	ua+5	ack	ig	ig	6
ig	ack	ig	6	ack	ig	obs	8
ig	ig	talk	ut	ig	talk	talk	ut+2
ig	ig	ack	ua	ig	talk	ack	ua+4
ig	ig	ig	10	ig	talk	ig	3
other combinations			0	ig	talk	obs	10
				ig	ack	talk	ut
				ig	ack	ack	ua+4
				ig	ack	ig	6
				ig	ack	obs	8
				ig	ig	talk	ut
				ig	ig	ack	ua
				ig	ig	ig	10
				ig	ig	obs	4
				other combinations			0

Figure 6.19: Network to decide the action to be taken by A with B. In the CPTs, F: A's friendliness with B, H: if A is in haste, D: how much A desires to talk to or greet a friend, Int: the character A's intended action, which can take on the values of talk (stopping to chat), ack (acknowledging) or ig (ignoring). The parameters f, h and desire[action to be taken] are character A's internal factors. In the utility table, Dis: if A is within a certain proximity of B, BI: interpretation of B's intended action with A, Dec: A's decision in its action with B, which can take on the values of talk, ack, ig, or obs (undecided as A is uncertain about the interpretation made about B), Util: the utility value which takes on the parameters ut and ua. The parameters ut and ua are weights for the talking and the acknowledging actions respectively, which are set based on what actions the character A has already taken, hence giving higher priority to keep A's action consistent.

P(AAS)					
na	su	sd	st	l	r
0.20	0.16	0.16	0.16	0.16	0.16



	P(SM)		P(CD)			P(CS)			
AAS	yes	no	l	r	nc	f	s	st	nc
na	0.50	0.50	0.10	0.10	0.80	0.10	0.10	0.10	0.70
su	0.90	0.10	0.20	0.20	0.60	0.90	0.00	0.05	0.05
sd	0.90	0.10	0.20	0.20	0.60	0.00	0.90	0.05	0.05
st	0.90	0.10	0.10	0.10	0.80	0.03	0.03	0.90	0.04
l	0.90	0.10	0.90	0.05	0.05	0.20	0.20	0.10	0.50
r	0.90	0.10	0.05	0.90	0.05	0.20	0.20	0.10	0.50

Figure 6.20: Network for interpreting adversary's avoidance strategy. The Conditional Probability Tables (CPTs) show the prior probabilities of adversary's avoidance strategy (AAS) and the probability of observed information given AAS. (na: no avoidance; su: speed up; sd: slow down; st: stopped; l: left; r: right; nc: no change; f: faster; s: slower).

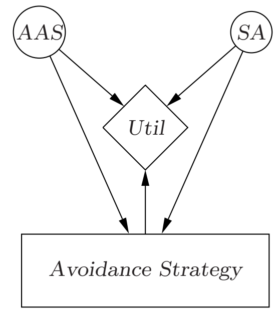
6.4 Collision Avoidance Behavior

In the previous sections, we have seen how our decision network framework can be applied to high level behaviors. We will now show that it is also readily applicable to low-level collision avoidance behaviors. Much behavioral animation research has focused on collision avoidance, but implementing a strategy that is capable of averting potential collisions from arbitrary directions within a confined space in a realistic, humanlike way remains a challenge. Our framework facilitates the incorporation of common sense into the collision avoidance strategy.

The character first tries to avoid any static obstacles. The avoidance strategy includes the options of steering to the left or right, or stopping when in imminent danger of collision. For dynamic obstacles, a collision potential value is computed for each obstacle and the one with the highest potential receives priority. In any given situation, the optimal strategy can be one of the following: no avoidance, speed up, slow down, stop, steer left, steer right, steer left with a speed change, steer right with a speed change, stop while steering left, and stop while steering right. The utility values are weights associated

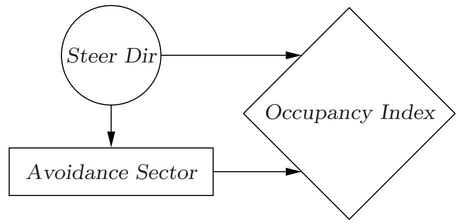
P(AAS)					
na	su	sd	st	l	r
0.20	0.16	0.16	0.16	0.16	0.16

P(SA=na)	P(SA=l)	P(SA=r)
0.40	0.30	0.30



SA=na			SA=l			SA=r		
AAS	AS	Util	AAS	AS	Util	AAS	AS	Util
su	na	wt[i]-0.1	na	na	-10	na	na	-10
su	su	f(wt[i]-0.3)	na	su	-10	na	su	-10
su	sd	wt[i]+0.1	na	sd	-10	na	sd	-10
sd	na	wt[i]-0.1	na	r	wt[i]-wt[l]	na	l	wt[i]-wt[r]
sd	su	wt[i]+0.1	na	sr	wt[i]-wt[l]	na	sl	wt[i]-wt[r]
sd	sd	f(wt[i]-0.3)	na	str	wt[i]-wt[l]	na	stl	wt[i]-wt[r]
st	na	-10	su	na	-10	su	na	-10
st	sd	wt[i]-0.1	su	su	-10	su	su	-10
st	st	-10	su	sd	-10	su	sd	-10
l	na	wt[i]+0.2	su	r	wt[i]-wt[l]	su	l	wt[i]-wt[r]
l	l	f(wt[i]-0.3)	su	sr	wt[i]-wt[l]	su	sl	wt[i]-wt[r]
l	sl	f(wt[i]-0.3)	su	str	wt[i]-wt[l]	su	stl	wt[i]-wt[r]
l	stl	f(wt[i]-0.3)	sd	na	-10	sd	na	-10
r	na	wt[i]+0.2	sd	su	-10	sd	su	-10
r	r	f(wt[i]-0.3)	sd	sd	-10	sd	sd	-10
r	sr	f(wt[i]-0.3)	sd	r	wt[i]-wt[l]	sd	l	wt[i]-wt[r]
r	str	f(wt[i]-0.3)	sd	sr	wt[i]-wt[l]	sd	sl	wt[i]-wt[r]
others		wt[i]	sd	str	wt[i]-wt[l]	sd	stl	wt[i]-wt[r]
			st	na	-10	st	na	-10
			st	su	-10	st	su	-10
			st	sd	-10	st	sd	-10
			st	st	-10	st	st	-10
			st	r	wt[i]-wt[l]	st	l	wt[i]-wt[r]
			st	sr	wt[i]-wt[l]	st	sl	wt[i]-wt[r]
			st	str	wt[i]-wt[l]	st	stl	wt[i]-wt[r]
			l	na	-10	l	na	-10
			l	su	-10	l	su	-10
			l	l	f(wt[i]-0.3)	l	l	-10
			l	r	wt[i]+0.1	l	r	wt[i]+0.1
			l	sl	f(wt[i]-0.3)	l	sl	-10
			l	sr	wt[i]+0.1	l	sr	wt[i]+0.1
			l	stl	f(wt[i]-0.3)	l	stl	-10
			l	str	wt[i]+0.1	l	str	wt[i]+0.1
			r	na	-10	r	na	-10
			r	su	-10	r	su	-10
			r	sd	-10	r	sd	-10
			r	l	wt[i]+0.1	r	l	wt[i]+0.1
			r	r	-10	r	r	f(wt[i]-0.3)
			r	sl	wt[i]+0.1	r	sl	wt[i]+0.1
			r	sr	-10	r	sr	f(wt[i]-0.3)
			r	stl	wt[i]+0.1	r	stl	wt[i]+0.1
			r	str	-10	r	str	f(wt[i]-0.3)
			others		wt[i]	others		wt[i]

Figure 6.21: Avoidance strategy network. In the CPTs, the symbols AAS and SA stand for Adversary's Avoidance Strategy and Static Avoidance, respectively. In the utility table, the avoidance strategy (AS) can take on the values of no avoidance (na), speed up (su), slow down (sd), stop (st), steer left (l), steer right (r), steer left with a speed change (sl), steer right with a speed change (sr), stop while steering left (stl), and stop while steering right (str). The parameter *wt[i]* is the weight associated with the corresponding avoidance strategy; for example, when *AS=sd*, for its utility value *wt[i]* is the weight associated with the slow down avoidance strategy. This weight parameter is determined based on the character's relationship with the adversary in terms of their position, orientation and speed, as well as if there are other characters and static obstacles around the character. The function f(x) for any given value of AS, say *as*, will return the value x if the character has previously decided to take avoidance strategy *as* or has not taken any action to avoid the obstacle, and will return the value -10 otherwise to signal an inappropriate avoidance strategy.



(a)

SD	P(SD)
l	.5
r	.5

(b)

```
utility(sector_weight[i],boundary_sec)
  if ((SD==l) and (boundary_sec > i)) or
      ((SD==r) and (boundary_sec < i)) then
    utility = -100
  else if boundary_sec = i then
    utility = -sector_weight[i] + bonus
  else
    utility = -sector_weight[i]
```

(c)

Figure 6.22: (a) Network for determining avoidance angle sector. (b) Prior probabilities of steering direction (l: left, r: right). (c) Occupancy index utility function to assess the desirability of each sector for the current avoidance strategy; the one with the lowest utility value will be chosen (sector_weight[i] is the weight value for sector i based on how crowded sector i is, boundary_sec is the sector corresponding to the closest possible avoidance direction; it depends on the location of the obstacle). A small constant bonus is subtracted to give a slightly higher priority to the closest avoidance direction.

with these avoidance strategy options. The weights are determined by our observations of how real humans balance avoidance choices under different circumstances. Due to the limitations of the DI-Guy virtual human motor system, a speed change command has a delayed effect, thus we currently do not use the speed change options, but have incorporated them in case a better motor system becomes available.

To avoid a decision network that is too large and intractable, we devise a hierarchical network structure to specify the collision avoidance strategy. A decision is made through the following series of decision networks. First the network depicted in Figure 6.20 is used to infer the dynamic obstacle character B's intended avoidance strategy of character A, based on A's perception of whether or not B sees it; if B is perceived to be changing direction or changing speed. Then the decision network depicted in Figure 6.21 is used to decide an avoidance strategy based on the inferred adversary's avoidance strategy, the static avoidance decision made earlier, and the utility values. Finally, should the decision in the previous step be to steer, the decision network in Figure 6.22 is used to determine the steering angle based on the sectoral occupancy index values.

Chapter 7

Behavioral Modeling: Additional Details

This chapter continues the presentation in Chapter 6, presenting the full details of our four decision network behavior models. The subsections parallel those in the previous chapter, but are presented bottom-up, in reverse order, beginning with the most recently discussed collision avoidance behavior and concluding with the emergency response behavior.

7.1 Collision Avoidance Behavior Model Details

The environment model provided by Shao [2005] provides the location of static obstacles and path planning information. The smallest angle away from the character's current heading direction that is free of static obstacle and close to the goal direction is chosen to avoid a static obstacle. This decision, which can lead to steering towards the left or right, is taken into account when the character must avoid dynamic obstacles.

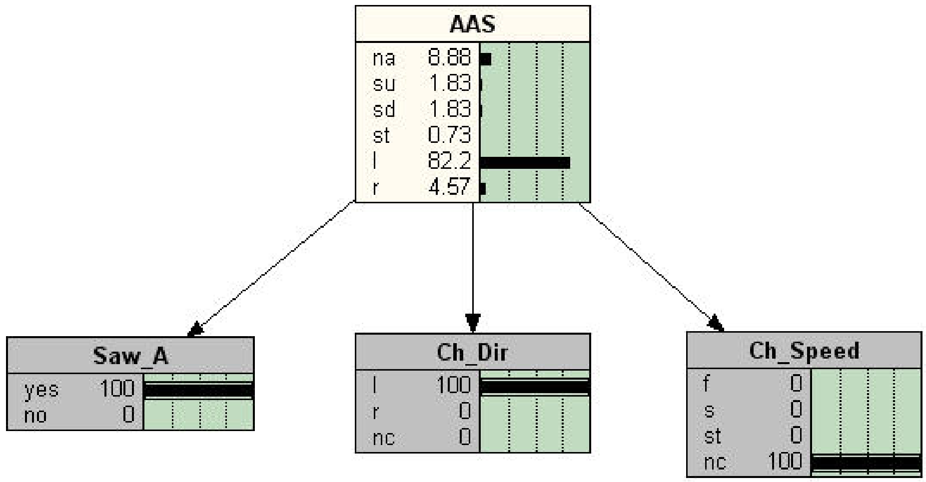


Figure 7.1: A sample belief-bar diagram generated for the network in Figure 6.20 to interpret an adversary's avoidance strategy. This illustrates the case in which the observation is entered into the network that the adversary has noticed the character, has changed its direction towards the left, and has not changed its speed. The network concludes that the adversary is turning left to avoid A, as state l has the highest probability.

7.1.1 Procedure Details

For each dynamic obstacle within character A's field of view (Figure 5.2), A first determines if it might collide with the obstacle. This is determined based on the position, speed, and orientation relationship between A and the corresponding obstacle. Within its field of view, character A also observes each dynamic obstacle's speed, orientation and gaze direction for two consecutive time steps. For each dynamic obstacle that may potentially collide with A, character A compares its observation of the adversary in the current and previous time steps in order to judge if this potential adversary has noticed character A, the adversary has no direction change or is steering left or is steering right, and if it has increased its speed, decreased its speed, has stopped, or has no speed change. These assessments are then entered as evidence in the network shown in Figure 6.20. Figure 7.1 shows a sample belief-bar diagram generated for the network in Figure 6.20. The diagram illustrates the case for which the observation is entered into the network that the adversary has noticed A, has changed its direction towards the left, and has not changed its speed. The interpretation made by the network concludes that the adversary is turning left to avoid A, as the state of the corresponding node has the highest *a posteriori* probability.

Once the adversary's intended avoidance strategy is assessed, a collision potential value is calculated based on the interpretation, the proximity of the adversary, and the proximity of the anticipated collision. The closer the adversary and the anticipated collision, the higher the potential value. The collision potential will be higher if the adversary is interpreted to make no avoidance attempt or has stopped, in which case A has to take avoidance measures. For persistence, a small bonus value is associated with the adversary that A was trying to avoid in the previous step. The dynamic obstacle with the highest collision potential value is further processed for avoidance strategy determination.

Given the existing steering direction associated with static obstacle avoidance and the assessment made about the dynamic obstacle with the highest collision potential, the network depicted in Figure 6.21 determines the character's final avoidance strategy. The utility table values for this decision depend on the weight parameter *wt[i]* where *i* is the index for the possible avoidance strategy, which can be as follows: no avoidance, speed up, slow down, stop, steer left, steer right, steer left with a speed change, steer right with a speed change, stop while steering left and stop while steering right. The weights are set in accordance with commonsense considerations about how people normally avoid collisions. They are initialized to 0. Preferred options are given positive values, with higher values for greater preference. Given the delayed speed change of the underling motor system, the speed up, slow down, steer left with a speed change, and steer right with a speed change options are not used; hence, their weights remain 0. The remaining weights are set based on Algorithm 1 in which *wst* is the weight for stopping, *wl* is the weight for steering left, *wr* is the weight for steering right, *wstl* is the weight for stopping while steering left, and *wstr* is the weight for stopping while steering right.

The strategy with the highest utility value will be chosen. Figure 7.3 shows a sample belief-bar diagram for this network to illustrate the case when the adversary's avoidance

Algorithm 1 Weight settings for avoidance strategies

```
wst=0; wl = 0; wr=0; wstl = 0; wstr = 0;
if (static avoidance strategy = steer left) then
  wl = 1.0
if (avoidance strategy = steer right) then
  wr = 1.0
if (character is very close to a non-stopping obstacle) then
  wst = 1.5
if (number of obstacles to left > right) then
  wr += 0.1
else
  wl += 0.1
if (very close to static obstacle on the left and have right side free) then
  wl -= 1.0; wr += 0.2;
else if (very close to static obstacle on the right and have left side free) then
  wr -= 1.0; wl += 0.2;
else if (both sides are tight) then
  wst += 2.0;
if (character is running into a head-on collision case) then
  weight for the non-obstacle side += 0.2;
if (obstacle is a distance away and is going to cross character's path as illustrated in 7.2(a)) then
  weight for the obstacle side += 0.5;
if (obstacle is a distance away and is going to cross character's path as illustrated in 7.2(b)) then
  weight for the non-obstacle side += 0.5; (to avoid colliding with the obstacle)
if (close to dynamic obstacle) then
  weight for non-obstacle side += 0.2; weight for obstacle side -= 2.0;
if (one side is heavily crowded judging by having high sectoral occupancy index values to one side) then
  weight for tight side -= 1.0; weight for free side += 0.1;
else if (both sides are heavily crowded) then
  wst += 3.0;
if (neither side is heavily crowded) then
  weight for less crowded side += 0.2;
if (neither side is tight) then
  weight for previous avoidance direction += 0.6; (to maintain consistency)
weight for direction closer to goal += 0.2;
if (direction closer to goal is not heavily crowded) then
  weight for direction closer to goal += 0.2; (favor goal direction even more)
if (character is now avoiding a different obstacle from the previous timestep) then
  weight for direction closer to goal += 0.3;
    (avoid turning further and further away from goal direction when need to avoid one dynamic obstacle after another)
if (character is turning over 90° away from the goal direction) then
  weight for direction closer to goal += 0.5; (avoid going in opposite direction from the goal direction)
if (wl = wr) then
  if (character has been avoiding collision since last timestep) then
    weight for previous collision avoidance direction += 0.1; (to break the tie)
  else
    weight for direction to the goal += 0.2;
if (character is stopped and very close to obstacle ahead) then
  exchange values for wstl and wl; exchange values for wstr and wr; (favor stop and steering direction instead)
```

strategy is interpreted to be turning towards the character's left and the character plans to turn right to avoid a static obstacle. With the weight parameter values shown in the diagram, the avoidance strategy of turning right has the highest utility value and hence will be chosen.

If steering is required in the decision, the network in Figure 6.22 is used to determine the avoidance sector. The weight for sector i, sector_weight[i], used in the utility function

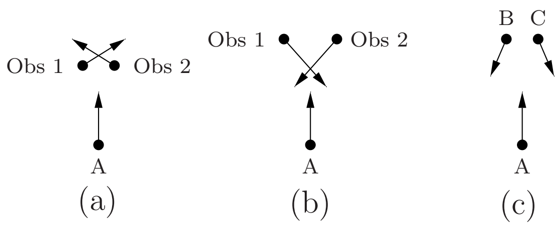


Figure 7.2: (a) Character A anticipates collision with either obstacle 1 or 2 when A catches up and the obstacle crosses its path (the arrows indicate the walking direction). In order to avoid colliding with the obstacle, the character should turn away from the current obstacle. (b) Character A anticipates collision with either obstacle 1 or 2 as they cross each other's path. In order to avoid colliding with the obstacle, the character should turn to the non-obstacle side. (c) Character A is not likely to collide with characters whose position and direction are similar to B or C.

Algorithm 2 Weight settings for perceptual sectors

```
sector_weight[0]=0; sector_weight[1]=-0.5;
sector_weight[2]=-1.0; sector_weight[3]=-1.5;
sector_weight[4]=-1.0; sector_weight[5]=-0.5; sector_weight[6]=0;
(these initializations are made to give increasingly higher preference to sectors with smaller angles away from the
character's current direction. The smaller the weight value, the higher the preference)
for (each character in view) do
  if (walking ahead of the character and is going in similar direction) then
    if (distance extremely close) then
      weight for the sector in which that character locates sector_weight[i]+=10;
    else if (distance quite far) then
      sector_weight[i]+=0.2;
    else
      sector_weight[i]+=0.1;
  if (walking towards the character) then
    if (distance extremely close) then
      sector_weight[i]+=10;
    else if (distance quite far) then
      sector_weight[i]+=0.2;
    else
      if (is not likely to collide with the character as illustrated in Figure 7.2(c)) then
        sector_weight[i]+=(distance_threshold-distance to the character)*0.2;
        (in this case, weight depends on distance from the character)
      else
        sector_weight[i]+=(distance_threshold-distance to the character)*0.2+(direction difference)*0.02;
        (in this case, also consider collision potentials)
for (each static obstacle ahead) do
  if (distance extremely close) then
    all sectors in which the obstacle occupies sector_weight[i]+=10;
  else if (distance quite far) then
    sector_weight[i]+=0.1;
  else
    sector_weight[i]+=0.5;
```

is determined by Algorithm 2.

The sector with the lowest sector weight value is the least crowded one. Since we pick the state with the highest utility value, we use the negated sector weight values in our utility table in the network. The sector with the highest utility value is associated with

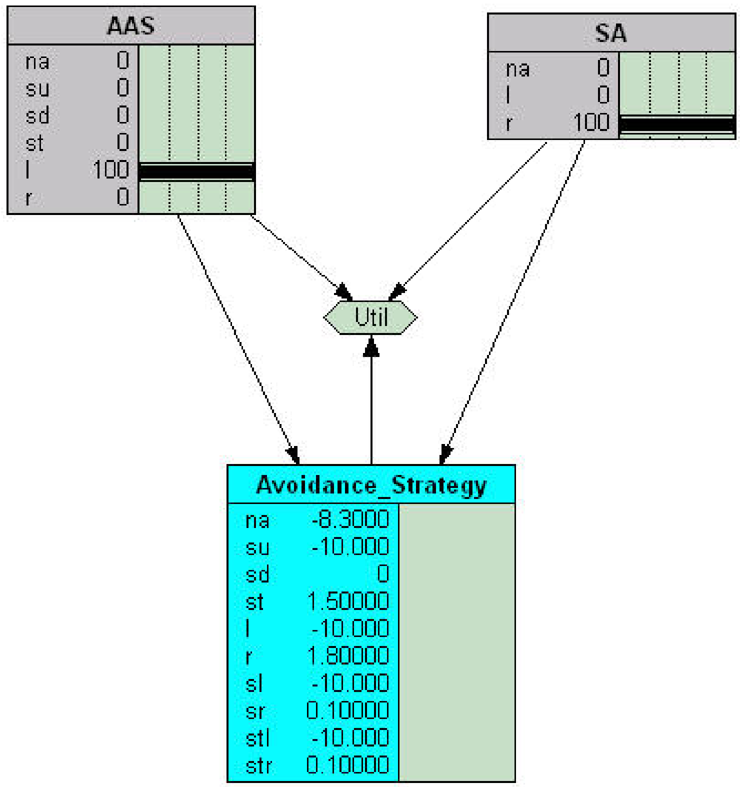
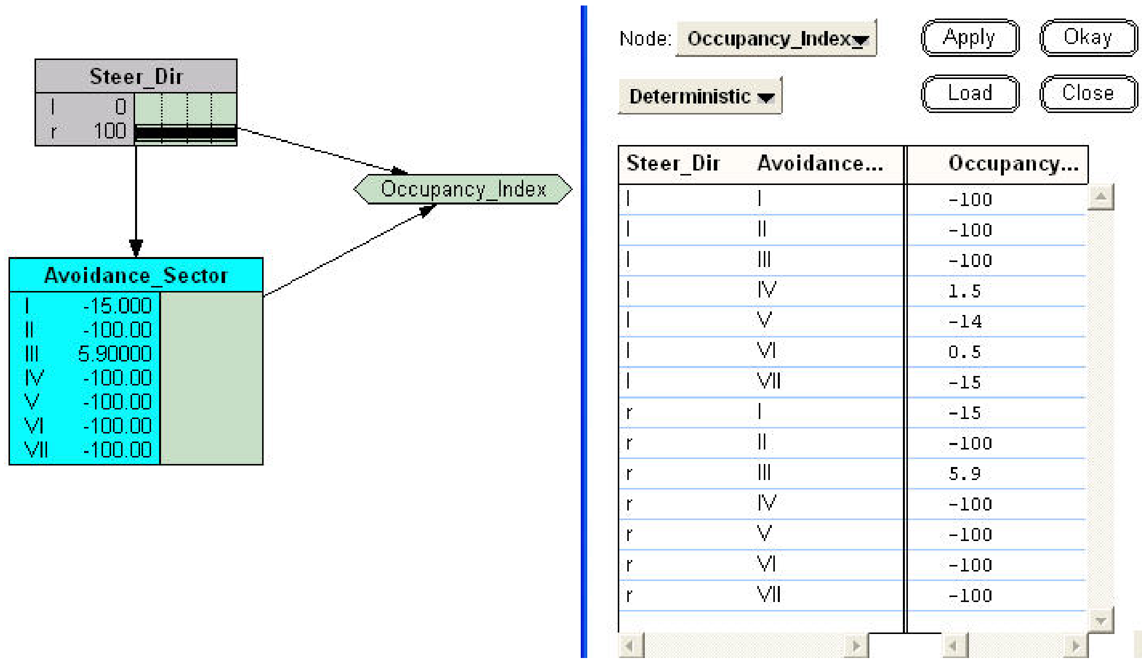


Figure 7.3: A sample belief-bar diagram generated for the avoidance strategy network in Figure 6.21.

the best steering direction, and hence it will be chosen as the avoidance angle to take. Figure 7.4 shows the belief-bar diagram, where the steering direction is turning left, and the sectoral weight values are as shown in the Avoidance_Sector table. Sector III yields the highest utility value and hence will be chosen as the preferred steering direction to avoid the obstacles.

7.1.2 Prior Probability and Utility Settings

In assessing the adversary's avoidance strategy (Figure 6.21), all the possible strategies are equally likely before any observation is made, with the "no avoidance" strategy being given slightly higher prior probability, as this will be the default strategy if no evidence can be collected. When the adversary character B is actively avoiding character A, the implication is that B must have noticed A; hence, we assign a high probability of 0.9



Steer_Dir	Avoidance...	Occupancy...
l	I	-100
l	II	-100
l	III	-100
l	IV	1.5
l	V	-14
l	VI	0.5
l	VII	-15
r	I	-15
r	II	-100
r	III	5.9
r	IV	-100
r	V	-100
r	VI	-100
r	VII	-100

Figure 7.4: A sample belief-bar diagram generated for the network in Figure 6.22 to determine avoidance angle sector.

to the *yes* state of the conditional probability for Saw_A given that B is taking any avoidance action. When B is not taking any avoidance action, it is possible that B has not yet noticed A. It is also possible that B saw A, but chose not to take any action. Hence, the conditional probability for Saw_A given AAS=na is assigned 0.5 for both states. As for the observation of a change in direction, unless the adversary B is making a direction change to avoid A, character A will most likely see no direction change in B. Similarly for a speed change, unless B's avoidance strategy involves a change in speed, A is less likely to observe any speed change in B. The conditional probabilities for these two chance nodes are assigned in order to reflect this.

For the network in Figure 6.21, the weight associated with each potential avoidance strategy *wt[i]* already captures the character's preference based on how crowded A's path is, without considering the interpreted adversary's avoidance strategy or its own planned static avoidance direction. The utility settings in Figure 6.21 follow these preferences, except for the cases where the combination of evidence for AAS and SA makes it necessary to make an adjustment. For example, a low utility value of -10 is assigned to make sure that impossible combinations are precluded.[1] For combinations where turning in a particular direction contradicts evidence observed about AAS and SA to some extent, the weight for this direction minus that for the opposite direction is set as the utility value. Thus, unless the weight for this proposed direction outweighs twice the weight for the other direction, the corresponding option will not be selected; for example, steering towards the left when B is interpreted as not taking any avoidance action and when A needs to turn right to avoid a static obstacle. There are also cases in which the avoidance option is generally inappropriate, but if the character has already been taking this strategy since the previous time steps, in order to maintain consistency, this option will be given the benefit of the doubt by assigning the utility function $f(wt[i] - 0.3)$, where the definition of the function f is given in the caption of Figure 6.21. If the combination of the AAS and SA observations makes the avoidance option less preferred, 0.1 is deducted from the weight preference $wt[i]$. For example, when there is no need to avoid any static obstacle and the adversary is perceived to be speeding up, choosing to take no avoidance action belongs to this category. For cases in which the combination of the AAS and SA observations makes the avoidance option preferable, a small constant ranging from 0.1 to 0.3 is added to $wt[i]$. For example, when the adversary is perceived to be speeding up, then slowing down is the preferred action.

The steering direction chosen by the network in Figure 6.21 will be given as evidence for the Steer Dir variable (SD) in the network of Figure 6.22. To avoid collision, character A will then try to find a turning angle that is as small as possible. In order to turn away from the obstacle, A needs to take the direction to the obstacle as the reference angle when searching for the appropriate steering angle. In this network, the state with the

[1] Examples of impossible combinations are when B is stopped and A proposes not to take any avoidance action, which will result in A crashing into B, or when A needs to turn left to avoid a static obstacle while B is turning to A's left in order to avoid A and A proposes to speed up, so that A will collide with the static obstacle sooner.

highest utility value will be chosen; hence, the negated values of the sector weights are used in the utility table. The situation becomes complicated in a crowded space where turning a certain angle will help the character successfully avoid the obstacle, but will make it vulnerable to collisions with another characters walking nearby. In such a case, the character needs to widen its turning angle to avoid both characters. The sector weights give the character a sense of how crowded its surroundings are. The higher the weight value, the more crowded the corresponding sector is. Given the sector containing the obstacle, the smallest turning angle will be the one in the same sector; hence, a small constant is added in the utility table to the sector weight for the corresponding sector in order to give it a slightly higher preference. The sectors in the opposite direction of the intended steering direction are obviously inappropriate; hence, they are ruled out by giving them a low utility value of -100.

7.2 Acquaintance Behavior Model Details

When a character encounters another character that it recognizes, the acquaintance behavior model is invoked to simulate interaction between these two characters. In acquaintance behavior, it is important for each character to interpret the other character's intended action. In real life, the interpretation is aided by numerous observable cues, such as body language, gestures, facial expressions, etc. Unfortunately, because of the limitations of the DI-Guy API, only a limited number of cues can be observed by our characters. Our framework is general, however, and it can easily incorporate additional observable traits should they become available. This simply requires the addition in the network of a chance node for each trait variable, its associated CPT, and corresponding links to indicate its relationship with existing nodes in the network.

7.2.1 Procedure Details

The three possible actions in our current acquaintance behavior model (Section 6.3) are (i) the two characters talk with each other, (ii) acknowledge each other without stopping to chat, or (iii) ignore each other.

First character A must assess the action in which the other character B is engaged. For example, if B is avoiding an obstacle, A cannot easily interpret B's intended action with A during obstacle avoidance, so A can only refer to the previous interpretation that it made before B starts to avoid the collision. Otherwise, if no such interpretation had been made, A must wait until B finishes collision avoidance to assess B's intention. If B is meeting other characters, A will give up its attempt to interact with B, since B is preoccupied.

In assessing if B is in collision avoidance, two cases must be considered. The first is when B initiates collision avoidance, in which case, B would be changing directions, and possibly also changing speed. The second is when B is in the midst of avoiding a collision, in which case, B may have already finished turning towards its intended steering direction, and hence does not appear to be changing its direction or speed from the previous time step. In the absence of these two cues, A can refer to the previous interpretation it has made about B to see if B is engaged in collision avoidance. For both cases, an important indicator will be whether B has any potential obstacle in its path.

Figure 7.5 shows a belief-bar diagram illustrating the case when there is no change detected in the direction or speed of B, but B appears to be dealing with obstacles, the network determines that B is not starting to avoid a collision. Given A's previous interpretation that B is in collision avoidance, the network assesses that B is not starting to avoid collision, but is already engaged in collision avoidance. The judgment made in this case depends heavily on the previous interpretation, since almost all other relevant cues are absent. To avoid locking into an interpretation in the absence of other relevant cues, the previous interpretation value is set to *yes* the first time A detects that B is in

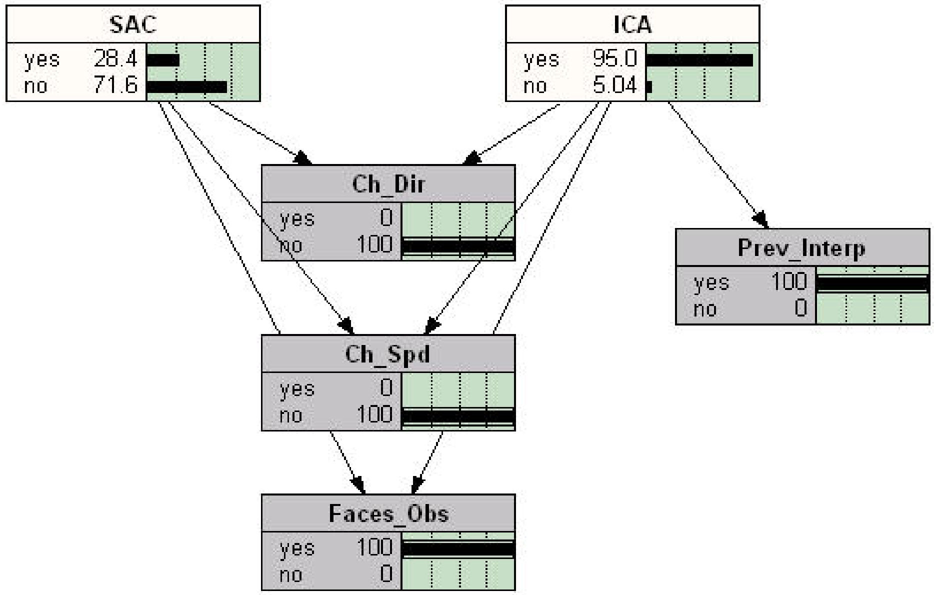


Figure 7.5: A sample belief-bar diagram generated for the network in Figure 6.15 to assess if the other character is engaged in collision avoidance.

collision avoidance, and this setting will expire after a time limit, during which all other relevant cues remain absent, then the previous interpretation value will be reset to *no*. By that time, if A still sees no change in speed or direction in B, then B is likely to have finished avoiding the collision, and A will interpret B as no longer being engaged in collision avoidance.

When starting to meet a third party, character C, character B is likely to change direction as it turns towards C, and to look at C and slow down. If B is in the process of walking towards C with the intention of chatting with C, B will be walking directly towards C. This is captured by the chance node Dir in the network in Figure 6.16. B will also be slowing down and looking at C. If B is already chatting with character C, B will be stationary and facing C within a certain proximity (which is captured by the chance node Face_Agt). Both cases belong to the category that B is in the process of meeting another character C, which the previous interpretation made about B should have indicated already. The belief-bar diagram in Figure 7.6 shows the case when all relevant cues are absent. A will conclude that B is not starting to meet another character C,

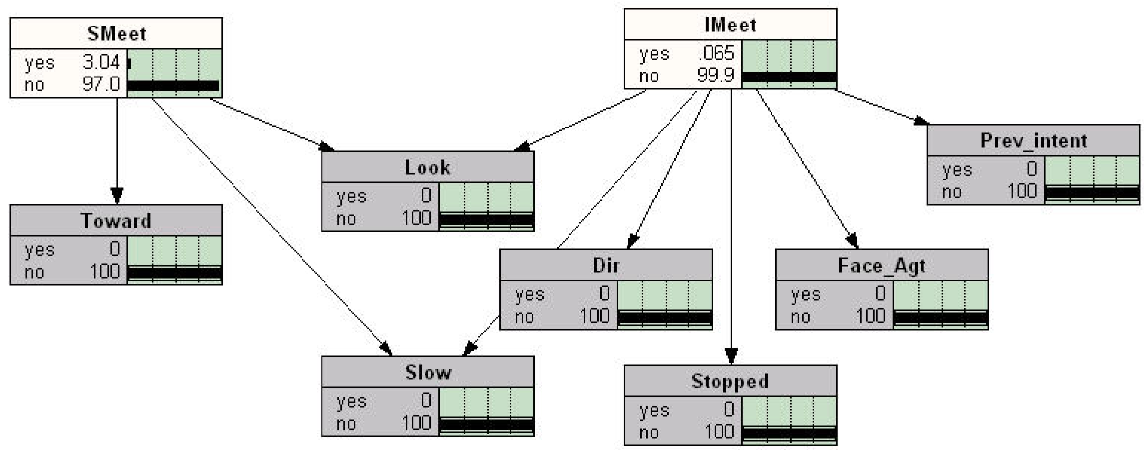


Figure 7.6: A sample belief-bar diagram generated for the network in Figure 6.16 that assesses if the other character is meeting a third party.

nor is it in the process of meeting C. The interpretations resulting from the networks in Figure 6.15 and Figure 6.16 are then entered as evidence into the network of Figure 6.17. The belief-bar diagram in Figure 7.7 illustrates the final interpretation A makes about B, i.e., that B is engaged in collision avoidance, when the interpretations obtained from the networks in Figure 7.5 and Figure 7.6 are entered as evidence. Hence, A will refer to the previous interpretation it made about B's intended interaction with it or wait for the next time step to reassess the situation.

If B is perceived not to be engaged in any particular action at the moment, A will continue its assessment of B's intention using the network in Figure 6.18, which evaluates whether B intends to talk to A, acknowledge A without chatting, or ignore A. This assessment is mainly based on observing cues, which would indicate that B intends to stop chatting with A. In the absence of these cues, A will try to determine if B is going to greet A without stopping. If B has the intention of talking to A, it may be just staring to walk towards A or could already be approaching A. These two stages exhibit different traits. When starting to walk towards A, character B should have at least seen A, and be turning towards A, which is captured by the chance node Toward_A in the network of Figure 6.18. While in the process of approaching A, character B will have finished turning

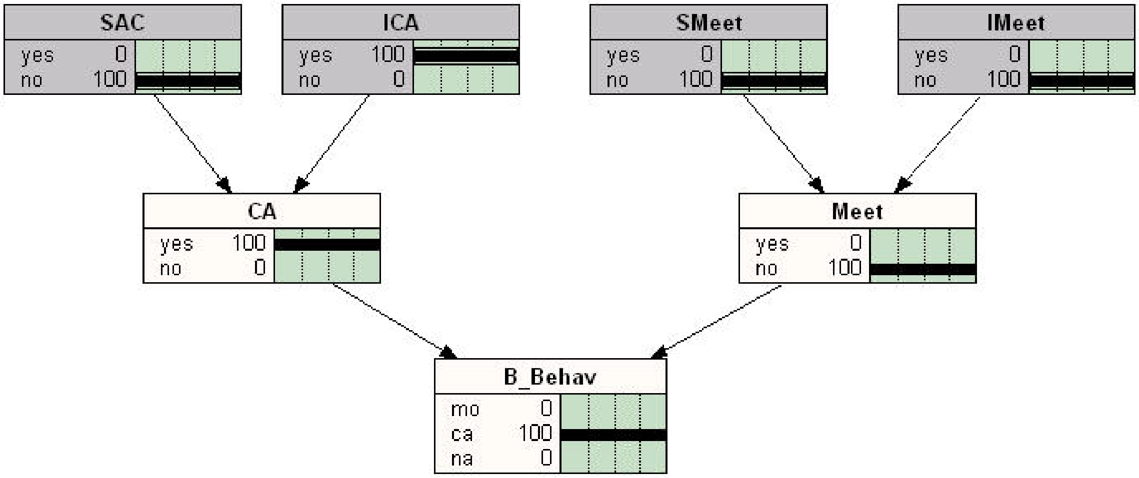


Figure 7.7: A sample belief-bar diagram generated for the network in Figure 6.17 that interprets the behavior in which the other character is engaged.

and is walking directly towards A (captured by the node Dir_to_A), but only assuming that the previous interpretation already showed a trace of such intention (captured by the node Prev intent). Both cases rely on whether B is looking at A and slowing down.

Figure 7.8 shows the belief-bar diagram for this network given evidence that B has seen A, B is not turning towards A, B is looking at A, B is not slowing down, B is walking towards A and the previous interpretation of B's intention was to talk. The posterior probabilities in the network indicate that B intends to talk to A and is already in the process of approaching A. As for any given node, the state with the highest posterior probability is taken as the chosen state.

After inferring B's intent, A must consider its own intention in order to decide how to interact with B. As shown in Figure 6.19, A's own intention is based on how friendly A feels about B, whether A is in a hurry, and the strength of A's desire to talk to somebody or just acknowledge a friend. There is uncertainty in the interpretation about B's intention because of the lack of or inaccuracy of the observed cues. While A and B are separated by a certain distance, A may choose to continue making observations to ensure it has not misinterpret B's intention; hence, the decision node for B's intent includes observation as one of its possible states. However once A and B are closer than a certain

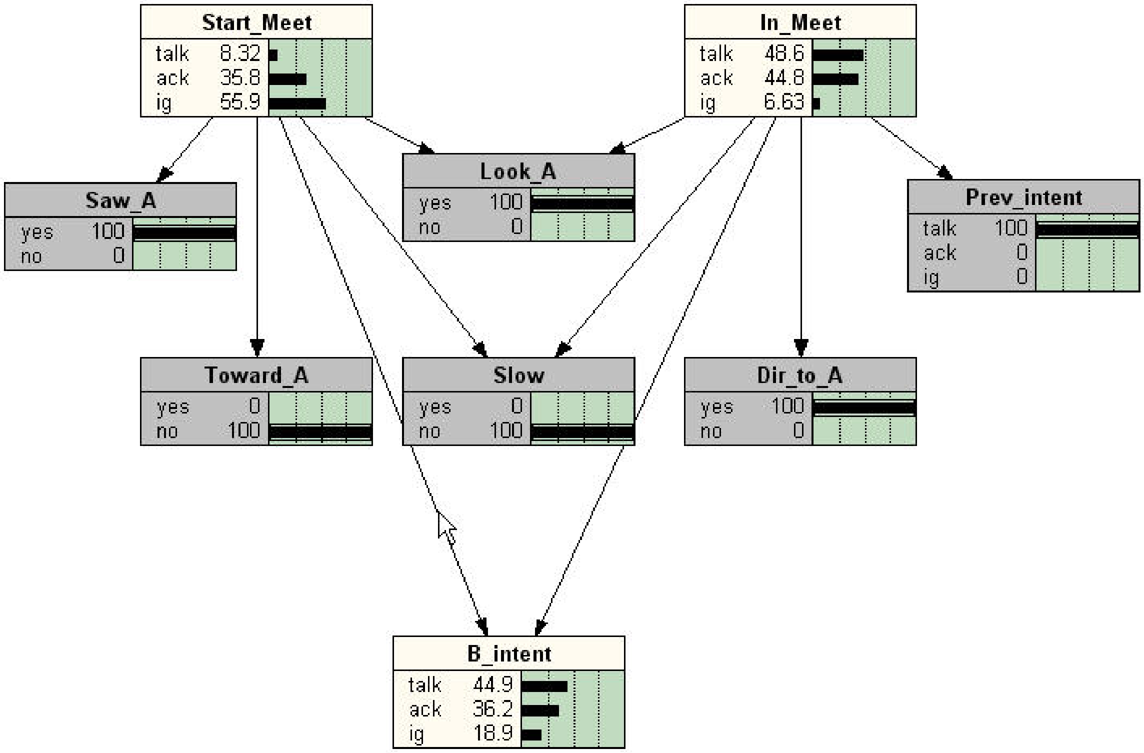


Figure 7.8: A sample belief-bar diagram generated for the network in Figure 6.18 to interpret the other character B's intended action with A.

distance, A will have to decide, otherwise it will walk past B without having expressed its intended interaction with B. The belief-bar diagram in Figure 7.9 illustrates that with B's intent interpreted earlier entered as evidence, A's internal friendliness towards B, whether it is in a hurry, its desire to interact with another character, and whether the distance between A and B is within the separation threshold, the decision state with the highest utility value, talk with B, is chosen. For this example, the parameters ut and ua are both 0. Since A's intended action is to talk with B and it perceives B as having the same intention, talking is the natural decision.

Should A's intention differ from its inference of B's intent, a tradeoff must be made. How this is done depends on the utility values, which will be explained in the next section.

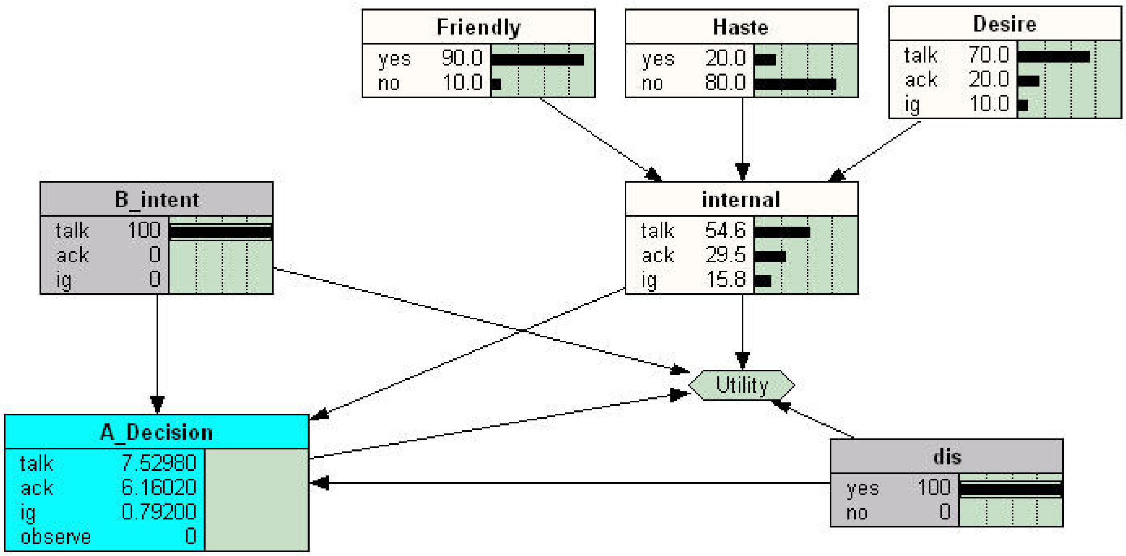


Figure 7.9: A sample belief-bar diagram generated for the network in Figure 6.19 to decide the action to be taken by A with B.

7.2.2 Prior Probability and Utility Settings

For nodes such as SAC, ICA, SMeet, IMeet, Start Meet, and In Meet, which represent data to be interpreted, there is initially no evidence to support any particular state, so their prior probabilities are equally divided among their possible states, with the exception of Start Meet and In Meet, which have three possible states, talk, acknowledgement and ignore, necessitating that a slightly higher prior probability be assigned to the *ignore* state, which is the assumed state in the absence of any observations. The conditional probabilities are assigned based on how the behavior designer chooses to weigh these variables. For SAC, when ICA is in the *no* state, the more indicative factor is whether character B has any potential obstacle with which it could collide. Other relevant factors are whether B is changing direction or speed, which often serve as cues that B is starting to avoid a collision. However, the absence of these cues is not as indicative, so they have been assigned a conditional probability of 0.5 when both SAC and ICA are in the *no* state. For ICA, the more indicative factor is the previous interpretation of B's behavior. When the character already starts to avoid the collision, whether B still has any potential obstacle with which it could collide becomes less indicative as it is possible that with the avoidance strategy B is taking, it is no longer in danger of colliding with any other character. Hence, the conditional probability for this variable given that the character is already in collision avoidance is 0.5. After B starts to avoid collision, before it finishes turning and changing speed, it is possible that both SAC and ICA will be interpreted as being positive. In this case, the variables Ch_Dir, Ch_Spd and Face Obs are not as indicative; hence, they are assigned a conditional probability of 0.5. After an initial set of prior probabilities has been assigned, the designer can use the Netica GUI to enter various evidence combinations and observe the resulting interpretations or decisions produced by the networks and make adjustments if necessary.

When it is just starting to meet another character, B is likely to be turning towards that character, looking at it, and slowing down. When it is already in the process of meeting another character, B is more likely to be looking at that character. When it is not meeting the other character, B is not likely to be looking at the character. When B just starts to avoid another character, it is more likely to slow down compared to when it is already meeting another character, as it may have already stopped to have a chat. The conditional probabilities reflect these scenarios. For IMeet, the most indicative cue is the previous interpretation being made. Other relevant factors include whether B is directed towards a character it may be meeting with, B has stopped, and B is facing a character it may be chatting with in close proximity. When B is not meeting others, these cues are likely to be absent.

Once the assessment of SAC and ICA are done, inferring whether B is in collision avoidance is made simple, because when either SAC or ICA is true, B is perceived to be in collision avoidance, otherwise B is perceived not to be preoccupied with collision avoidance. Similarly the inference of whether B is meeting another character is made based on the SMeet and IMeet assessments made earlier. The behavior in which B is engaged is deduced from these two assessments. When both CA and Meet are true, this indicates an inaccurate interpretation, in which case B is assumed to be in collision avoidance, since in this case A would refer to the previous interpretation it made or try to observe again and make another assessment at the next time step.

The process of assessing A's intention toward B (Figure 6.18) is similar to assessing if B is meeting a third party, except that instead of just monitoring two states, *yes* and *no*, we are now concerned with three possible states: talking with A, acknowledging A, or ignoring A. The conditional probabilities are assigned based on the following observations. When B wants only to acknowledge A, B is less likely to be walking towards A or be oriented towards A. It is also less likely to be slowing down, but B should still have noticed A and be looking at A. The variable Prev intent is more indicative than other factors affecting the assessment for In Meet.

Once A finishes inferring B's intention, it needs to decide, based on its own intention, how to interact with B (Figure 6.19). A's intended action with B depends on how friendly it feels towards B, whether A is in a hurry, and A's intrinsic desire to interact with others. Among these factors, A's desire plays a key role. However, in the case when A tends to ignore others yet feels very friendly towards B, A may still intend to acknowledge B. When A has a strong desire to talk with others, but is in a hurry and does not feel so friendly towards B, it may intend only to acknowledge B or ignore B. The conditional probabilities for the chance node Internal reflect these scenarios.

In view of the uncertainties involved in the assessment of B's intention, when A and B are separated by a certain distance, A has the time to verify its interpretations by continuing to observe B. The observing action is denoted by the *obs* state in the decision node. Once A and B approach a certain separation threshold, A must make a decision about its interaction with B. Hence, in this case, the *obs* state is assigned a utility value of 0 in order to rule it out. The final decision is the state with the highest utility value. To maintain consistency, the parameters ut and ua for the utility table are set based on the action that the character A has already taken. Should A's previous decision be to talk or acknowledge, ut and ua are set to be 5 and 5 respectively, otherwise they are assigned a value of 0. The bonus values ut and ua are set to 0 when the previous intention was to ignore, since the ignoring action means that character A has not yet started to interact with B as of last time step. Should the decision at this time step be to start talking or acknowledging B, then A can start that action without having to worry about keeping consistency with any interaction strategy used in the previous time step. Hence, no bonus value is needed in this case.

The constants added in the utility table are used to specify preference depending on A's intention, B's inferred intention, and the distance factor. When B's inferred intention matches A's, the corresponding interaction state is given a bonus of 10 in order to make it the preferred action to take. When the distance threshold has not yet been met, and B's intent differs from A's own intention, a relatively high utility value is assigned for the obs state in order to allow A to make further observations and try to decide later. When the distance threshold is reached, the basic scheme for assigning preference is that when B's intent is to talk, A will follow its own intention more closely. When B's intent is not to talk, then even if A wants to talk, it will not be productive to do so, so the talk state will be given lower preference than other states.

7.2.3 Personality Simulation

The network of Figure 6.19 is used in the acquaintance model to decide the action to be taken when character A encounters another known character B. How friendly A feels about B and whether A is in a hurry affects how A might want to acquaint with B. A's extroversion personality trait determines how talkative A is and how much A wants to be around people. This is reflected in the desire variable that indicates the strength of A's desire to greet B in the corresponding way. Table 7.1 shows that when the observation of B's intent and the Distance variable remain the same, different settings for the personality and internal factors alone will lead to different decisions, analogous to real humans. For

Friendly	Haste	Desire	Internal	A's Decision
yes	yes	talk	talk	talk
yes	yes	ack	ack	ack
yes	yes	ig	ack	ack
yes	no	talk	talk	talk
yes	no	ack	ack	ack
yes	no	ig	ig	ack
no	yes	talk	ack	ack
no	yes	ack	ack	ack
no	yes	ig	ig	ig
no	no	talk	ack	ack
no	no	ack	ack	ack
no	no	ig	ig	ack

Table 7.1: These data are generated by the network of Figure 6.19 when B_intent is *talk* and dis takes the value of *yes*.

example, when A thinks that B intends to chat, and they are approaching one another, A chooses to ignore B when it dislikes B, is in a hurry, and has low desire to greet anyone; however, when A has a low desire to greet anyone, A will acknowledge B when A feels friendly towards B and is not in a hurry.

7.3 Partnering Behavior Model Details

When two friends going in opposite directions see one another, they would normally approach each other and chat prior to deciding whether to proceed in partnership. Hence, when a character sees another character that it recognizes approaching, it will first employ the acquaintance model to decide how to interact with the character. Should they decide to have a chat, while talking they may decide to form a partnership. Another possible scenario is when a character sees another character walking ahead, if it has an interest in forming a partnership with that character, it would have to decide whether to make an effort to catch up to that character and make a partnering request. We will focus on the latter in explaining our model.

Haste	Flexible	Social	Friend	Desire	Dominant	Goal Match	Internal	Join
yes	yes	yes	yes	yes	yes	yes	yes	yes
yes	yes	yes	yes	yes	no	yes	yes	yes
yes	yes	yes	yes	no	yes	yes	no	no
yes	yes	yes	yes	no	no	yes	no	no
yes	yes	yes	no	yes	yes	yes	no	no
yes	yes	yes	no	yes	no	yes	no	no
yes	yes	yes	no	no	yes	yes	no	no
yes	yes	yes	no	no	no	yes	no	no
yes	yes	no	yes	yes	yes	yes	yes	yes
yes	yes	no	yes	yes	no	yes	yes	yes
yes	yes	no	yes	no	yes	yes	no	no
yes	yes	no	yes	no	no	yes	no	no
yes	yes	no	no	yes	yes	yes	no	no
yes	yes	no	no	yes	no	yes	no	no
yes	yes	no	no	no	yes	yes	no	no
yes	yes	no	no	no	no	yes	no	no
yes	no	yes	yes	yes	yes	no	yes	yes
yes	no	yes	yes	yes	no	no	yes	yes
yes	no	yes	yes	no	yes	no	no	no
yes	no	yes	yes	no	no	no	no	no
yes	no	yes	no	yes	yes	no	no	no
yes	no	yes	no	yes	no	no	no	no
yes	no	yes	no	no	yes	no	no	no
yes	no	yes	no	no	no	no	no	no
yes	no	no	yes	yes	yes	no	yes	yes
yes	no	no	yes	yes	no	no	yes	yes
yes	no	no	yes	no	yes	no	no	no
yes	no	no	yes	no	no	no	no	no
yes	no	no	no	yes	yes	no	no	no
yes	no	no	no	yes	no	no	no	no
yes	no	no	no	no	yes	no	no	no
yes	no	no	no	no	no	no	no	no
no	yes	yes	yes	yes	yes	yes	yes	yes
no	yes	yes	yes	yes	no	yes	yes	yes
no	yes	yes	yes	no	yes	yes	no	no
no	yes	yes	yes	no	no	yes	no	no
no	yes	yes	no	yes	yes	yes	no	no
no	yes	yes	no	yes	no	yes	no	no
no	yes	yes	no	no	yes	yes	no	no
no	yes	yes	no	no	no	yes	no	no

Table 7.2: A's decision to form a partnership with B affected by personality and internal factors. This data is generated by the network of Figure 6.11 when Goal Dir is set to be yes, and Perception assumes the state of unknown.

7.3.1 Procedure Details

The two main considerations character A makes before deciding to catch up to character B and initiate a partnering request are its own intention and whether it feels that its goal is compatible with that of the potential partner B. When A's goal direction and B's walking direction are consistent, then A is likely to feel there is a compatibility. When A is not in a hurry and is fairly flexible about changing its current goal direction to accommodate its potential partner's, it may feel there is a compatibility even if its goal direction differs significantly from that of B's walking direction. A's intention depends on how social it is, how friendly it feels about B and how much it desires to pair up with someone. When A's intention is to form a partnership with B but does not feel its goal is compatible with that of B's, a dominant character is less likely to bend and initiate the request. The perception in Figure 6.11 denotes A's perception of B's intention. When B is walking ahead, A cannot tell B's intention yet, so it will take on the state of unknown for this case. When A and B meet fact to face, this judgment usually comes from their conversation.

Figure 7.10 shows the belief-bar diagram for the network in Figure 6.11. The observation that B's walking direction is within 45° of A's goal direction is taken to mean that their goal directions agree, and this information is entered as evidence for the Goal Dir node. A's internal factors are entered as prior probabilities for the Haste, Flexible, Social, Friend, Desire, and Dominant nodes. As B is walking ahead, the state of the Perception node is unknown. As shown in this figure, A feels that its goal is compatible with that of B and it intends to form a partnership, so the final decision is the state with the highest utility value, which is to join B.

Once A decides to join B, it will speed up and walk towards B. When it approaches B, it will initiate a partnering request. B then must consider how it feels about this request. Since the initiating request comes from A, character B need not be concerned about goal compatibility, but need only consider its own desire and internal factors. There are two

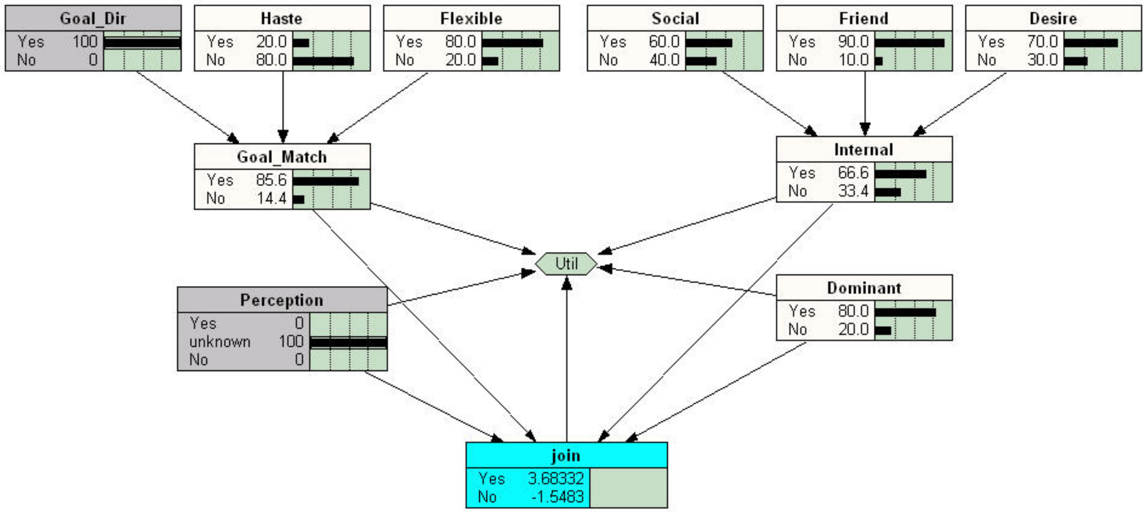


Figure 7.10: A sample belief-bar diagram generated for the network in Figure 6.11 to decide whether to form a partnership.

more factors B must take into consideration: one is the length of time that A has been making the request. If B had initially rejected the advance, but A repeated the request, B may be persuaded to change its mind. The other factor is how persistent B's personality is. B may be persuaded to change its mind more easily when A continues repeating its advance request and when B is not so persistent in its initial decision. An example set of parameters are given in Figure 7.11. As illustrated in this belief-bar diagram, given a low desire in forming a partnership, B rejects A's advance even though it feels quite friendly towards A.

Once B decides to reject the advance, it would continue walking. A must observe B's reaction to interpret B's response. The motor system provides only limited cues that A can observe, so we included only three observable variables in the network in Figure 6.13. When B receives the partnering request, should it choose to accept, B would typically turn to look at A in order to talk with A and is likely to stop and chat with A before carrying on. B may also choose to turn to face A while talking, which is a less reliable indicator than the previous two. The reliability node is added in the network to represent this. The belief-bar diagram shown in Figure 7.12 demonstrates that after having entered

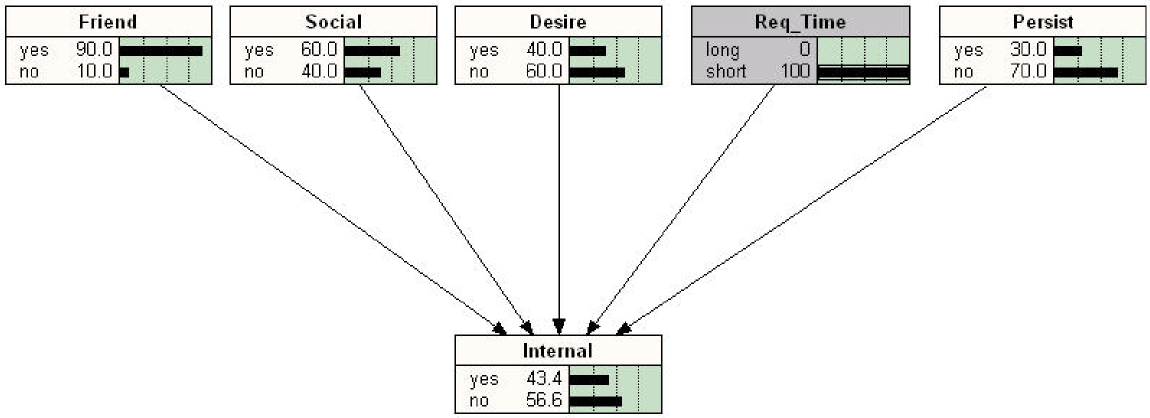


Figure 7.11: A sample belief-bar diagram generated for the network in Figure 6.12 to assess whether to accept a partnering request.

as evidence the observation that B is looking at A, but has not stopped or turned to face A, A concludes that B has rejected its request.

How A responds to the rejection depends on how well A can take rejection, how persistent it is in pursuing its own goal, and the strength of its intention to form a partnership, as well as how long it had been attempting to make the request. The value for the chance node Internal comes from the corresponding node in the network of Figure 6.11 after interpretation. The observation that the time A has spent in making the request has not yet exceeded a threshold has been entered as evidence. A's internal factors of being able to take rejection well and being quite persistent have been entered as the prior probabilities for the corresponding nodes. The assessment of the internal intention generated by the network in Figure 7.10 is entered as likelihood for the node Internal, since the posterior probabilities for this node in the network of Figure 7.10 indicate how inclined A is to form a partnership with B given the evidence presented. As a result, A decides it can tolerate the rejection and will continue its persuasion effort. Hence, there will be another cycle of A making the advance, B deciding whether to accept it, A interpreting B's reaction, and deciding what to do about it. This loop will end only when either B is persuaded to accept the request, in which case A and B will pair up, or when A gives up on B and they go their separate ways.

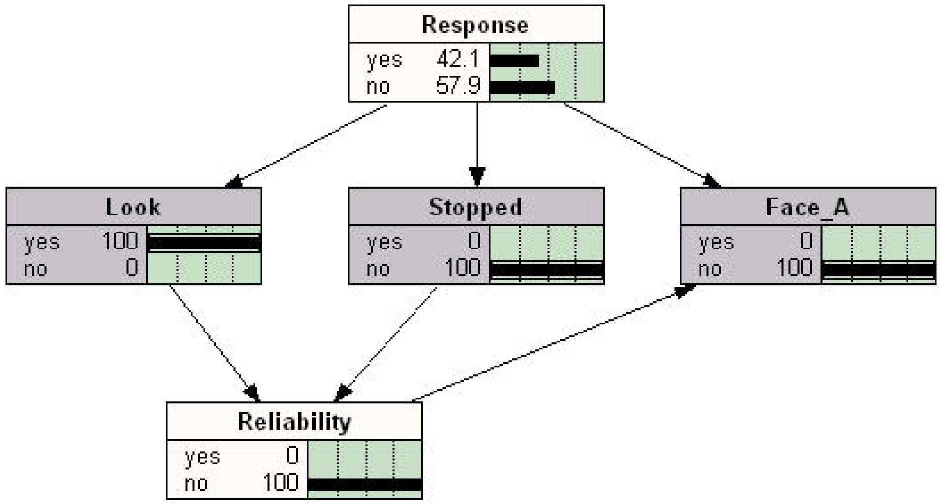


Figure 7.12: A sample belief-bar diagram generated for the network in Figure 6.13 to interpret potential partner's response to A's partnering request.

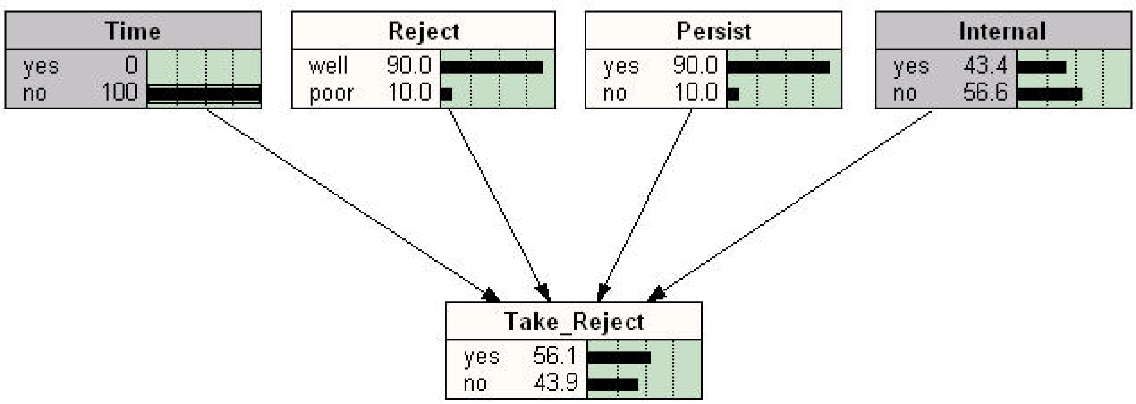


Figure 7.13: A sample belief-bar diagram generated for the network in Figure 6.14 to assess how to take rejection.

7.3.2 Prior Probability and Utility Settings

The character's personality and internal factors determine the prior probabilities for the nodes. The priors of unknown variables are set to make each possible state equi-probable. However, the unknown state in the Perception node in the network of Figure 6.11 is given a slightly higher probability as this is the assumed state when no evidence is available.

The conditional probabilities are set in an intuitive manner. In the network of Figure 6.11, for the variable Goal Match, the most favorable scenario is when A's goal direction is compatible with that of B and when A is not in a hurry and is fairly flexible about changing its current goal, hence $P(\text{GM} \,|\, \text{GD} = yes, H = no, Fl = yes)$ is set to 1.0 for GM = yes, and 0.0 for GM = no. The opposite is equally true: When A's goal does not match B's and when A is in a hurry and not flexible about changing its goal, then A is least likely to find a goal match; hence, $P(\text{GM} \,|\, \text{GD} = no, H = yes, Fl = no)$ is set to 1.0 for GM = no, and 0.0 for GM = yes. The remaining CPT entries are set somewhere in between these two extreme cases depending on how favorable the associated condition is for finding a goal match. Similarly for variable Internal, the most favorable condition is when A is social, feels friendly towards B, and has a desire to form a partnership, hence $P(Internal \,|\, S = yes, F = yes, D = yes)$ is set to 1.0 for $Internal = yes$, and 0.0 for $Internal = no$. The least favorable scenario is when A is not social, feels hostile towards B, and has no desire to form a partnership. In this case, $P(Internal \,|\, S = no, F = no, D = no)$ is set to 1.0 for $Internal = no$, and 0.0 for $Internal = yes$. The remaining CPT entries are set somewhere in between these two extreme cases, depending on how favorable the associated condition is for A to be inclined to form a partnership with B.

The utility table considers relevant factors including the Goal Match, the perception of B's reaction, A's internal inclination for partnering with B, and A's personality dominance. It specifies A's preference for each of the possible combinations of these contributing factors. The higher the utility value, the higher the preference. The most preferred combination of contributing factors for a given decision is given the highest utility value of 10, and the least preferred combination is given the lowest utility value of -10. For example, the combination when A is dominant, finds a goal match with B, intends to partner with B and B is perceived to want to join partnership too, is a preferred scenario for A to decide to partner with B; whereas the combination of the same conditions, but with A deciding to not to partner with B is considered one of the least likely combinations, and hence is assigned a utility value of -10. Similarly, other combinations are assigned utility values between -10 and 10 depending on how likely the behavior designer feels each combination would be.

In the network of Figure 6.12, the CPT entries for the Internal variable are set based on how likely the corresponding combinations of contributing factors would lead to B inclining to accept A's request to form a partnership. For example, the most likely combination is when B feels friendly towards A, is social, has the desire to form a partnership, A has requested to partner for some time, and B is not persistent, so in this case $P(Internal = yes \,|\, Friend = yes, S = yes, D = yes, RT = l, P = no)$ is set to 1.0. The combination of B feels hostile towards A, is not social, has no desire to form a partnership, A has not been requesting a partnership for long, and B is persistent, is one of the least likely combinations, and hence $P(Internal = yes \,|\, Friend = no, S = no, D = no, RT = s, P = yes)$ is set to 0.0. The remaining entries are assigned in a similar fashion.

In the network of Figure 6.13, the CPT entries for the variables Look and Stopped represent the likelihood that the corresponding variable state would be observed given the response. For example, if B's response is to accept A's request, it is more likely that B will be looking at A and will be stopped. Since B turning to face A is not as reliable an indicator as B looking at A and B stopping, the variable Reliability is added. When B accepts A's request, then B is less likely to exhibit the cue of turning to face A rather than stopping and looking at A. Character B can only turn to face A when B is stopped and looking at A; hence, the cue of facing A is only valid when the other two cues are present. We have added the reliability node to the network to reflect this. Hence the CPT entries for the Reliability variable are set such that the reliability value is set to *yes* only when $LA = yes$ and $ST = yes$, and is set to *no* in all other cases. The conditional probabilities for the variable Face A (FA) are set such that when Reliability is true, FA is more indicative and hence has higher probability of being present when B's response is *yes* than when Reliability is false.

For the network of Figure 6.14, a *yes* value for the Take Reject node means A will continue its persuasion effort by reiterating its partnering requests to B, whereas a *no* value means A will give up. We assume that A will give up when the time it has spent making unsuccessful advances has exceeded a time threshold. Hence, when $TT = yes$, $P(TR = yes)$ is set to 0.0, and $P(TR = no)$ is set to 1.0. When $TT = no$, the other factors Reject, Persist and Internal have an effect. The most likely internal factor combination for A to continue its effort is when A can take rejection well, is persistent, and really wants to form a partnership with B, hence $P(TR = yes \mid TT = no, R = yes, P = yes, Int = yes)$ is set to 1.0. The least likely combination is when A cannot take rejection well, is not persistent, and has a low desire to form a partnership with B; hence, $P(TR = yes \mid TT = no, R = no, P = no, Int = no)$ is set to 0.1. The remaining conditional probabilities are set somewhere in between these two extreme cases depending on how likely the behavior designer feels the corresponding combinations of internal factors would result in A wanting to continue making advances.

7.3.3 Personality Simulation

The partnering model uses the network of Figure 6.11 to determine if the character wants to form a partnership. In this decision process, the internal factors of whether the character is in a hurry and how flexible it is about its current goal, how friendly it feels towards this potential partner, and the strength of its desire to form a partnership with this potential partner are all factors that the character must consider. The social characteristic of the extroversion personality trait and the dominance characteristics of the character also play a role. Tables 7.2 and 7.3 demonstrate how these factors affect the decision making, all other factors remaining the same. Posterior probabilities evaluated by the network are compared, and the state with the highest posterior probability is taken as the decision. For illustration purposes in the table, instead of the posterior probability value, we record only the state with the highest posterior probability value

Haste	Flexible	Social	Friend	Desire	Dominant	Goal Match	Internal	Join
no	yes	no	yes	yes	yes	yes	yes	yes
no	yes	no	yes	yes	no	yes	yes	yes
no	yes	no	yes	no	yes	yes	no	no
no	yes	no	yes	no	no	yes	no	no
no	yes	no	no	yes	yes	yes	no	no
no	yes	no	no	yes	no	yes	no	no
no	yes	no	no	no	yes	yes	no	no
no	yes	no	no	no	no	yes	no	no
no	no	yes	yes	yes	yes	yes	yes	yes
no	no	yes	yes	yes	no	yes	yes	yes
no	no	yes	yes	no	yes	yes	no	no
no	no	yes	yes	no	no	yes	no	no
no	no	yes	no	yes	yes	yes	no	no
no	no	yes	no	yes	no	yes	no	no
no	no	yes	no	no	yes	yes	no	no
no	no	yes	no	no	no	yes	no	no
no	no	no	yes	yes	yes	yes	yes	yes
no	no	no	yes	yes	no	yes	yes	yes
no	no	no	yes	no	yes	yes	no	no
no	no	no	yes	no	no	yes	no	no
no	no	no	no	yes	yes	yes	no	no
no	no	no	no	yes	no	yes	no	no
no	no	no	no	no	yes	yes	no	no
no	no	no	no	no	no	yes	no	no

Table 7.3: A's decision to form a partnership with B affected by personality and internal factors (Continued).

for the corresponding node in order to demonstrate the final decision being made. For example, a social character who has the urge to form a partnership with someone, but is in a hurry and is not flexible in changing its current goal will not join another character who is going in the same direction but towards which it does not feel friendly; however, it will choose to form a partnership if it feels friendly towards the other character.

7.4 Emergency Response Behavior Model Details

In the event of an emergency, everyone who becomes aware of it will first perform a mental evaluation of the seriousness of the situation. One normally considers many factors in the

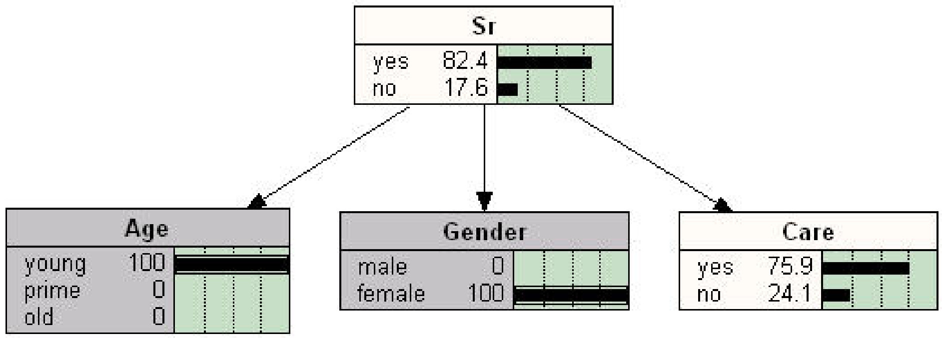


Figure 7.14: A sample belief-bar diagram generated for the network in Figure 6.2 to assess the seriousness of the situation.

assessment; however, in view of the limited cues that our characters can observe given the current underlying motor system, we have chosen to consider three traits, namely the age and gender of the patient and how caring the character is.

7.4.1 Procedure Details

Figure 7.14 shows a sample belief-bar diagram for the network in Figure 6.2, where the patient is a young female and the character is quite caring about others, hence the assessment is that the situation is fairly serious.

This evaluation is then entered as prior for the node Sr in the network of Figure 6.1, since the posterior probabilities for this node in the network of Figure 7.14 indicate the likelihood that the character believes the emergency situation is serious. The response of the character to the emergency depends on how altruistic the character is and its courage. An altruistic character is more inclined to investigate the situation and determine if there is anything that it can do to help. Even though it may want to help, a cowardly character may choose not to approach the scene for fear that it may be too horrific. The possible reactions that are simulated include ignoring, running over to check, and walking over to take a look at the scene. For the ignoring action, the character would turn its head to look in the direction of the emergency, but keeps walking on. Concerned characters will want to investigate the emergency scene. The more concerned will run over, while

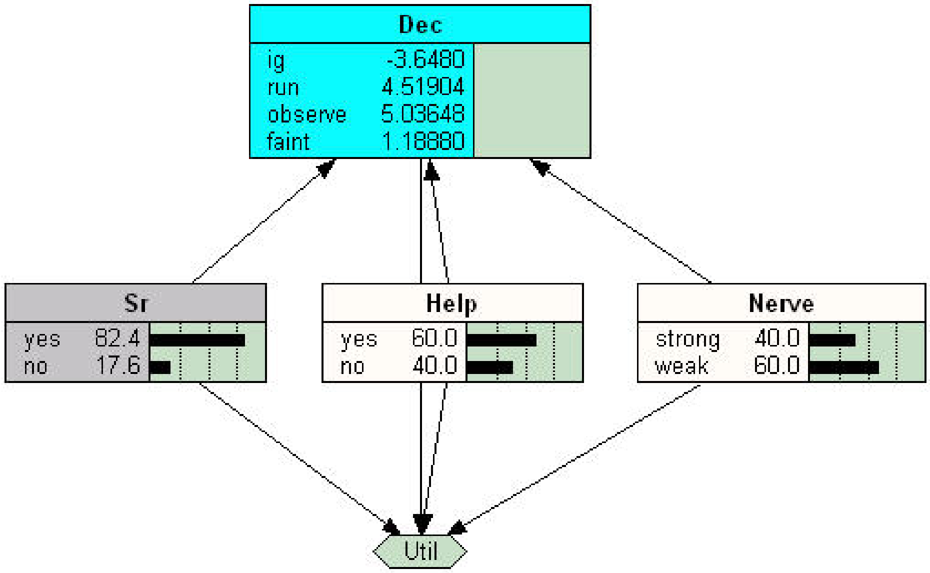


Figure 7.15: A sample belief-bar diagram generated for the network in Figure 6.1 to determine how to respond to the emergency.

the less concerned will walk over. The belief-bar diagram in Figure 7.15 shows that a relatively helpful character with less courage, who considers the situation to be quite serious, would choose to walk over to investigate the emergency scene.

Among the characters who are circling the emergency site and investigating the situation, the more resourceful ones may decide to fetch help. The possible actions at this stage include not fetching help, calling out to nearby people for help, and actively finding a law enforcement officer to inform them of the situation. This decision depends on how serious the character believes the situation is, how resourceful the character is in finding help, its observation whether there are other people already summoning help, and if the character had noticed a law enforcement officer previously and therefore knows a likely place to find one. Faced with a serious situation, a resourceful character may choose to find a police officer that it saw earlier if there is no one else fetching help. If it does not know the whereabouts of a law enforcement officer, it may also choose to call out to nearby characters for help. The belief-bar diagram in Figure 7.16 shows the case where a resourceful character, who believes that the situation is serious, and did not see anyone

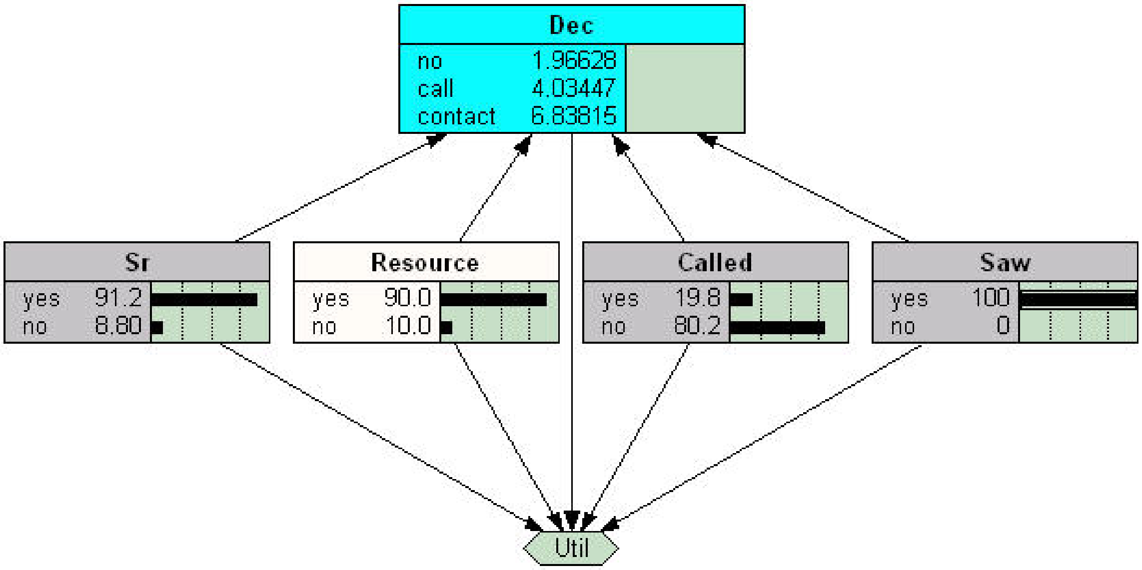


Figure 7.16: A sample belief-bar diagram generated for the network in Figure 6.3 to decide whether to fetch a police officer.

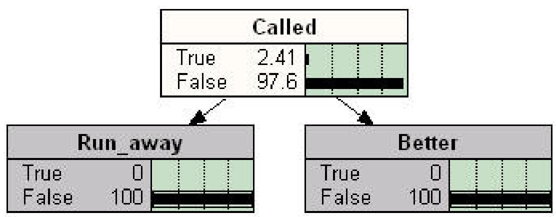


Figure 7.17: A sample belief-bar diagram generated for the network in Figure 6.4 to assess if someone else is summoning the police.

else fetching for help, but had seen a police officer earlier, decides to go to find the officer and request assistance.

The assessment of the seriousness of the situation results from the network of Figure 6.2, and the interpretation of whether someone else had called for help is made by the network of Figure 6.4. A person in the process of fetching help will typically be running away from the scene, hence this is included as a cue to be observed in making the interpretation. In the case where others appear to be calling for help, but the character believes that it can find a police officer sooner, the character may still decide to take action. The belief-bar diagram in Figure 7.17 illustrates the case where the character interprets that no help has been summoned yet, since it did not see any other character

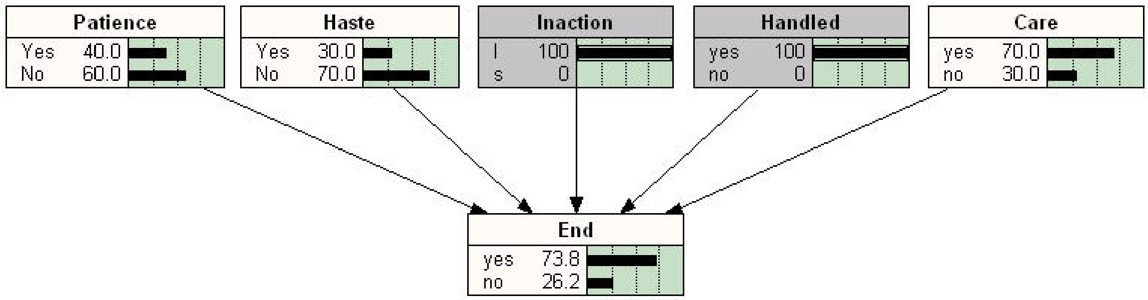


Figure 7.18: A sample belief-bar diagram generated for the network in Figure 6.5 to decide to end the emergency response behavior.

running away from the scene, and neither does there seem to be a character in a better position than itself to obtain assistance.

Characters investigating the emergency scene that have not gone to fetch assistance, may choose to leave the scene after a span of time during which they have not been able to do anything to help. The network of Figure 6.5 is used to make this decision. It looks at whether the character is patient, is in a hurry, how altruistic the character is, if someone else has taken action to handle the emergency, and how long the character has been standing by and unable to help. The belief-bar diagram in Figure 7.18 illustrates the case where with someone else already handling the emergency, a caring but relatively impatient character, who is not in a hurry but has been idling for a long time, has decided to end its emergency response behavior, and resume the pursuit of its original goal.

7.4.2 Prior Probability and Utility Settings

The prior probabilities of nodes representing the character's personality or internal factors are set according to the character's characteristics. The priors for unknown variables, such as the Sr node in the network of Figure 6.2 are set so as to give each state equal possibility, as initially there is no evidence favoring any particular state. Other conditional probabilities are set based on the behavior designer's intuitions.

In the network of Figure 6.2, the CPT entries for the nodes Age, Gender and Care are assigned based on the seriousness of the given emergency situation and how likely it is to observe the corresponding states of these variables. Assuming that when the situation is serious, the patient is most likely to be elderly, less likely to be a child, and least likely to be in the prime of age, the probability of $P(Age = o \,|\, Sr = yes)$ is set higher than $P(Age = y \,|\, Sr = yes)$, which in turn is set higher than $P(Age = p \,|\, Sr = yes)$. People tend to think that females are more vulnerable and consequently judge the situation to be more serious when a female patient is involved. So the probability $P(Gender = female \,|\, Sr = yes)$ is set much higher than $P(Gender = male \,|\, Sr = yes)$. An altruistic character will also tend to consider a situation more serious than a non-caring one, so the probability $P(Care = yes \,|\, Sr = yes)$ is set much higher than $P(Care = no \,|\, Sr = yes)$.

In deciding how to react, the assessment of the seriousness of an emergency, which is obtained from the network of Figure 6.2, is entered as prior for the node Sr in the network of Figure 6.1. We set the utility table to specify the character's preference over the possible combinations of relevant factors. The higher the utility value the higher the preference. The most unlikely combination, a helpful and courageous character choosing to ignore a serious emergency situation, is given a low utility value of -10. On the contrary, the same character choosing to run over and investigate is given the highest utility value of 8. Other utilities are set similarly, depending on how plausible the corresponding combination is.

For concerned characters who already approached and investigated the emergency scene, the network of Figure 6.3 enables them to decide whether they want to summon help. The utility table links the relevant factors with the decision and specifies the preference. The most favorable combination is given the highest utility value of 10, while the least favorable one is assigned 0. For example, when the situation is serious, and nobody has called for help yet, a resourceful character who saw a law enforcement officer a short while ago will most likely go contact the officer rather than take no action. Hence the utility for the former combination is assigned 10, whereas it is assigned 0 for the latter. Similarly if the character has not seen a police officer earlier, hence does not know where to find one, it is more likely to call out for help than to search for an officer; hence, the utility for the call action is assigned 10 instead.

In the network of Figure 6.4, when assessing if someone else had called for help, the CPT entries for the variable Run_away and Better are set based on how likely it is to observe the corresponding state given the condition of the Called variable. For example, when a character is seeking assistance, it is more likely to see that character running away from the emergency scene. It is also important for that character to be in a better position to get the required help. Since this interpretation will be used in the network of Figure 6.3 to decide whether to summon help, it is assumed for simplicity that, even if another character is attempting to contact police, if one is actually in a position to summon help more quickly, for example, by being closer to the officer or by being able to run faster, etc., the character will regard this as no help is being fetched. Hence the conditional probabilities of $P(B = yes \,|\, Called = yes)$ and $P(R = yes \,|\, Called = yes)$ are both given high values, while $P(B = yes \,|\, Called = no)$ and $P(R = yes \,|\, Called = no)$ are set low. Because even when someone is running away to get help, that person must also be in a better position than the character itself to get help in order for the character to interpret the situation as help being fetched, the probability of $P(B = yes \,|\, Called = yes)$ is given slightly higher value than $P(R = yes \,|\, Called = yes)$.

7.4.3 Personality Simulation

The agreeableness personality trait plays a role in the emergency response model, as an agreeable person tends to be more caring about others, so does the patience characteristic of the conscientiousness trait. A patient character may not mind lingering at the scene as much as an impatient character would. When a character tries to determine whether to end its own emergency response behavior and leave the accident site, internal factors such as whether the character is in a hurry also has an influence. Table 7.4 shows data

Patience	Haste	Care	End
yes	yes	yes	no
yes	yes	no	yes
yes	no	yes	no
yes	no	no	no
no	yes	yes	yes
no	yes	no	yes
no	no	yes	no
no	no	no	no

Table 7.4: This data are generated by the network of Figure 6.5 when Inaction takes the state of *short* and Handled is of value *no*.

obtained from the network of Figure 6.5 to illustrate how these factors affect the decision when no one had taken concrete action to handle the emergency yet and the character has not been present at the scene for long. For example, a character in a hurry who cares for others and is patient decides to stay longer, while a non-caring though patient character in a hurry chooses to leave.

Chapter 8

Results

Taking advantage of our decision network framework's ability to handle uncertainty and complexity, we have used it to equip our virtual human model with the aforementioned behavior models. As described earlier, the collision avoidance model incorporates anticipation of the opposing character's avoidance strategy and makes a decision accordingly. In addition to simulating the interpretations a character draws about its environment and about its potential partner's intentions, the acquaintance, partnership, and emergency response models also take into account the effect that personality and internal factors have on decision making. Decisions are made based on observations and the analysis of objects that fall within the character's focus of attention.

Our pedestrian animation system enables us to run simulations of autonomous, mutually interacting pedestrians in large-scale urban environment without human intervention. The automatically generated behavioral animations demonstrate the effectiveness of our decision network framework.

8.1 Pedestrians in Penn Station

We have used the Penn Station virtual environment model developed by Shao [2005] as the platform for our pedestrian simulation. There are two main areas in the station, the main waiting room and the concourse, where most activities take place. These two areas are connected through a passage. At run time, the environment model requires approximately 90MB of memory to accommodate the station and all the objects.

As in Shao [2005]'s system, we also use a text configuration file to initialize the pedestrians. Individual differences are reflected in these specifications. They include motion preference data such as preferred walking speed and maximal angular speed during turning, geometric appearance related data such as the character type and their dress style, and decision related personality traits and internal factors. The decision related parameters include whether the character is in a hurry, how social it is, its flexibility about the goal that it is pursuing, how dominant it is, if it is persistent, how well it can take rejections, how much it cares about others, how helpful it is, how courageous it is, how resourceful it is, and what other characters the character knows, and how friendly it feels towards each of them, in particular, its desire to greet, chat with, and form a partnership with someone.

We have populated the train station mostly with commuters plus a few law enforcement officers. Though our system is capable of generating crowd behaviors, we concentrate on simulating individual behaviors, more specifically detailed interactions between two characters. While acting autonomously, each character observes its surroundings, makes interpretations about its observations, and acts based on common sense rules and its own impression about nearby characters and other objects.

8.2 Control Flow in the Autonomous Characters

Each character acts autonomously inside the virtual environment. Let us follow a character, call it Jane, to illustrate how the system is integrated together. At any given time, Jane's intention generator processes her internal attributes and memory to assess what Jane's current intention is, for example, Jane may want to buy a ticket or go to Platform

3 to catch a train, or simply wonder around the train station to kill some time. Jane's eyes observe her environment and determine what objects are within Jane's 180-degree field of view through querying the database. This includes a list of static objects and a list of other characters within Jane's field of view. The perception data Jane can gather from querying the database includes position, speed and orientation of the objects in the environment, including other characters. Jane can also observe the gaze directions of other characters.

Jane's attention filter examines these sensory data and assesses what object(s) fall into her focus of attention. This depends on which objects are of interest to Jane's current intention, or if there is any object that makes a sudden, rapid move, as such an object tends to grab one's attention. When Jane sees in her focus of attention a character she recognizes, or a character she may have an interest to interact with, or when there is a high collision potential with some character, Jane draws inferences about them through her interpreter in her perception system. The interpreter applies the interpretation capability of our framework to make inferences about others' intentions; hence, Jane can decide how to interact with them.

Jane's decision maker takes in the interpreted perceptions, while taking into account Jane's current intention to make a decision. This decision making process is accomplished with our behavioral framework. Each character is equipped with the four behavioral models we have built, which were described in Chapters 6 and 7. These models are event triggered—i.e., whenever the circumstances give rise to certain behaviors, the corresponding networks are evaluated to make necessary decisions. For example, when Jane bumps into another known character, her acquaintance behavior model is invoked to help her decide how to greet her friend, whereas if she sees an emergency situation, her emergency response behavior becomes active. The networks associated with each behavior model are only invoked on a need-to-use basis.

Each character possesses a set of behavior routines that allow them to carry out primitive actions, such as walking to certain locations, stopping to chat, etc. The basic motions involved are provided by the motor system. Once an action selection decision is made, the corresponding behavior routines are called to carry out the necessary actions. For example, if Jane saw an emergency situation and decides to get help from a police officer she saw earlier, Jane's behavior routine will communicate with the motor system and allow Jane to run to the position where she last saw the police officer. In the meantime, if Jane detects any potential collision, her collision avoidance behavior model will be invoked to assess the potential obstacle's avoidance strategy and make an avoidance decision based on that. This decision may temporarily interrupt the running motion Jane was engaged in and cause her to take a detour before resuming calling the police. The motor system is responsible for carrying out the actual primitive movements such as walking and running.

As another example, when Jane decides to form a partnership with Lisa, who is walking ahead. The corresponding behavior routine will allow Jane to walk faster and catch up with Lisa. Then Jane will stop and make an advance request. If Lisa's partnering model decides to accept Jane's request, Lisa's behavior routine will make Lisa stop to look at Jane and they chat briefly before walking on together.

Our behavioral framework is applied mostly in the interpreter and the decision maker, while behavior routines to carry out the corresponding decisions are implemented to make the decisions realizable. The object that catches most of the character's attention, which is determined by the attention filter, is used to direct the character's gaze direction. When there is no particular object of interest, the character will look forward at where it is going.

8.3 Animation Samples

Our most complex animation is the "emergency response" animation illustrated in Figs. 8.1, 8.2 and 8.3. A young female commuter feels ill, staggers, and then faints to the floor of the concourse area in the virtual Penn Station. Some commuters around her turn their heads to look, but keep walking on. Others walk over to see what happened. The most concerned run over to take a closer look. After checking the patient, a woman commuter recalls seeing a law enforcement officer in the main waiting room of the station and, hoping that he is still around, decides to run over there and ask him to help. While she is gone, some of the commuters attending to the patient at the scene decide to leave as they determine that they cannot do much now that someone is already seeking assistance, while other passers-by approach to take a closer look at the scene. The law enforcement officer who was summoned by the woman eventually reaches the scene, examines the patient and decides to radio paramedics to provide further assistance.

Our shorter simulations demonstrate the functionality of the partnering behavior model, differing only by internal factor and personality settings. One animation (Figure 8.4) shows a pedestrian A who notices a friend B walking ahead, catches up, and tries to partner with B, who initially refuses the advance, but reconsiders upon A's insistence, and then permits A to partner and walk/talk together. A second animation (Figure 8.5) shows a male pedestrian eventually giving up on his attempts to partner with a female stranger, after repeatedly being shunned. The third animation (Figure 8.6) shows two female friends who unexpectedly encounter one another. They stop to have a chat. Then one of them decides to join the other for a walk since she has some spare time and nothing better to do.

8.4 Performance

Our decision network behavior framework is efficient. An autonomous pedestrian evaluates the relevant networks at each step of the animation only when an associated decision must be made. On an Intel Xeon 3.2GHz PC with 1GB RAM, the behavior system for a single pedestrian usually takes under 1msec to execute. The most frequently invoked networks are those for collision avoidance. The network depicted in Figure 6.21, which has the largest utility table, takes from 1–2msec to set up and compile in Netica and usually less than 1msec to evaluate. The emergency response network shown in Figure 6.1 takes about 1msec to compile and 1msec to evaluate.

The behavior models are event triggered, so these networks are invoked only when their corresponding decisions or interpretations need to be made. When the emergency response model is invoked, the total simulation time involved, including the network evaluation time, the execution time for the corresponding behavior routines, and the time for sensing the environment is about 10msec total. The network related simulation time is less than half of this. When no decision needs to be made, and the character simply carries out what it is doing, the simulation time is about 6msec.

(1)

(2)

(3)

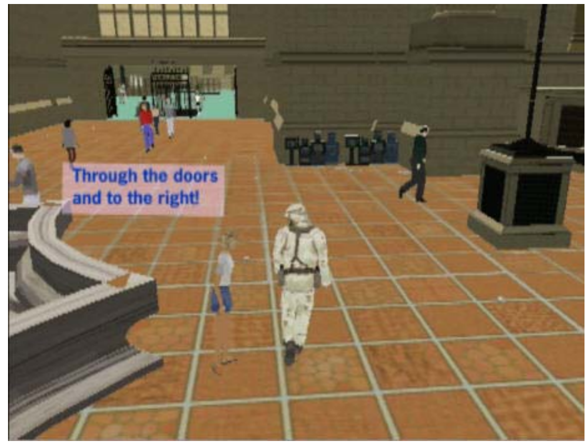
(4)

(5)

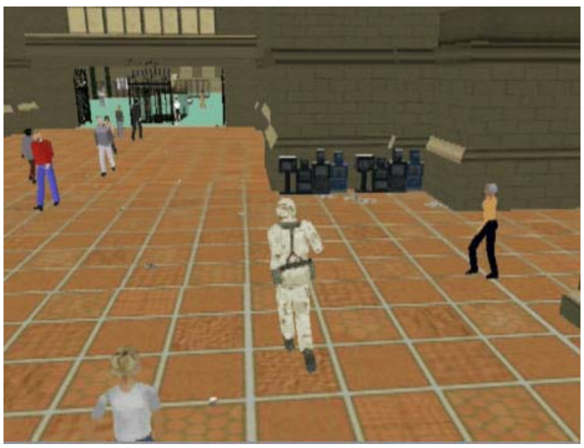
(6)

Figure 8.1: Emergency Response.

(7)

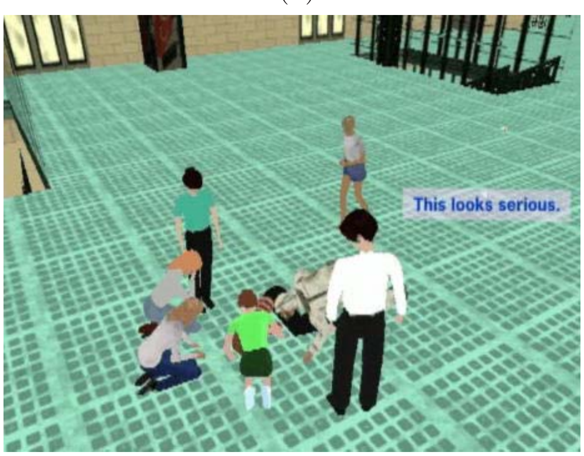


(8)



(9)

(10)



(11)

(12)

Figure 8.2: Emergency Response (continued).

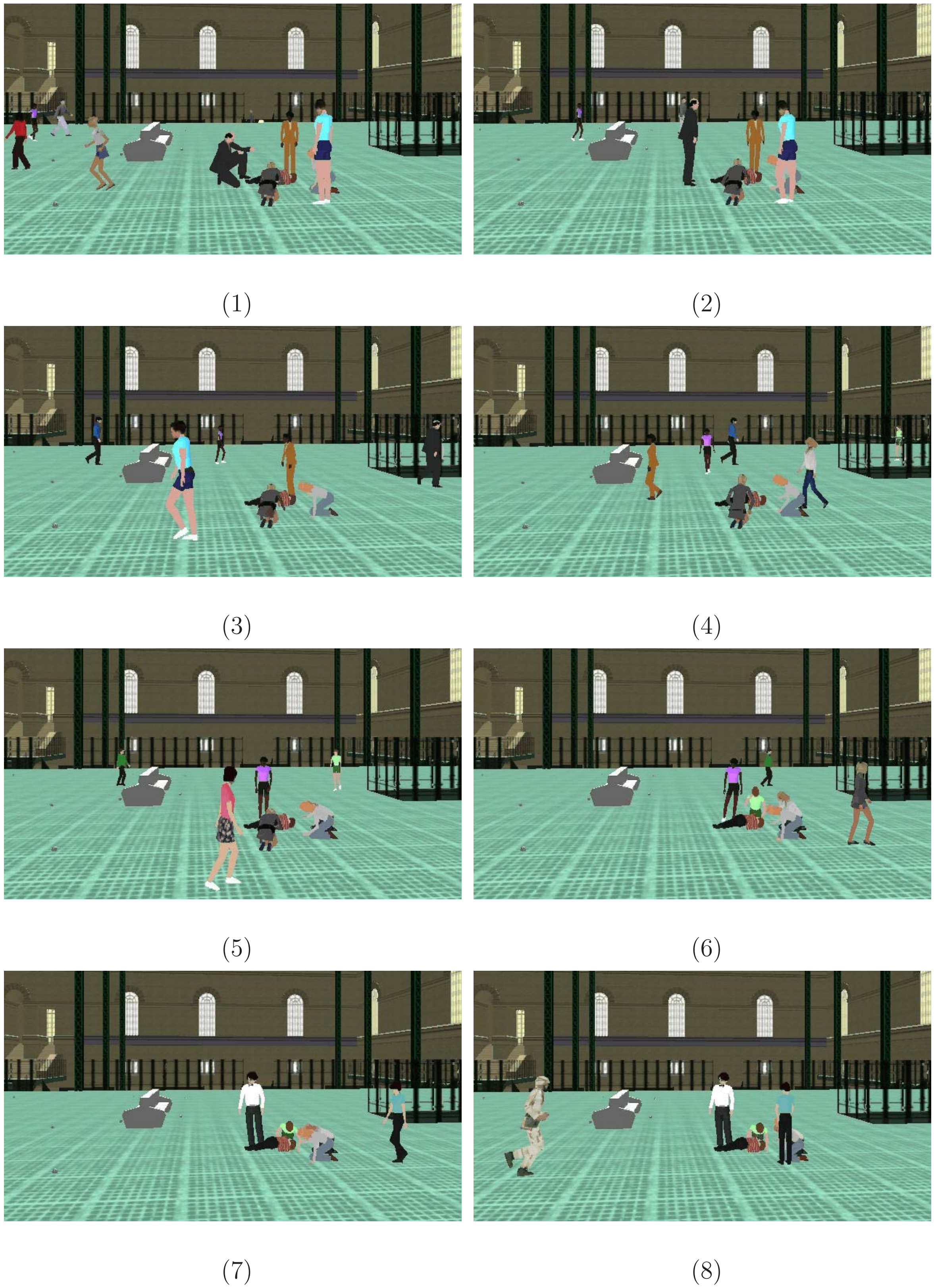
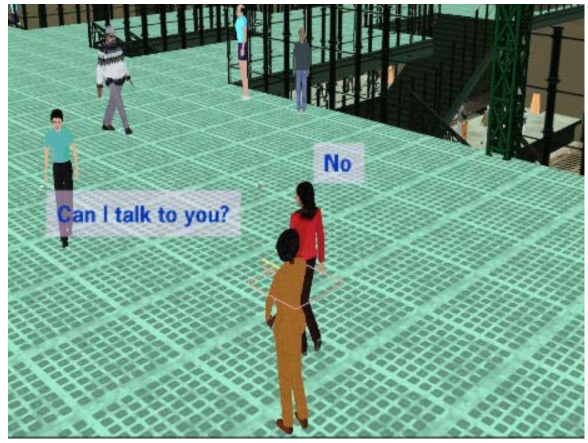


Figure 8.3: Overview of the emergency site while a character runs to fetch help.

(1)



(2)

(3)

(4)

(5)

(6)

Figure 8.4: Partnering behavior model: accepting the request.

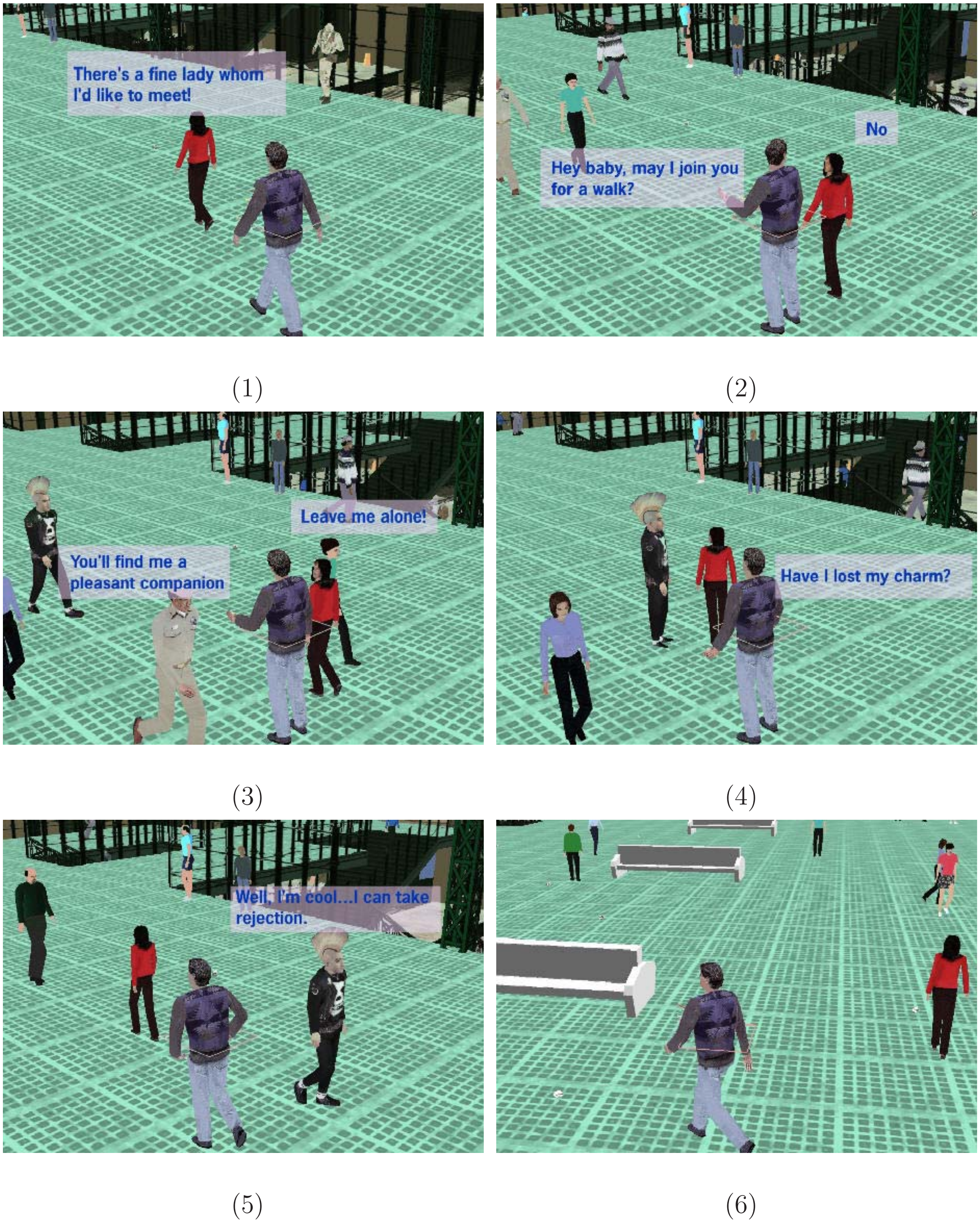
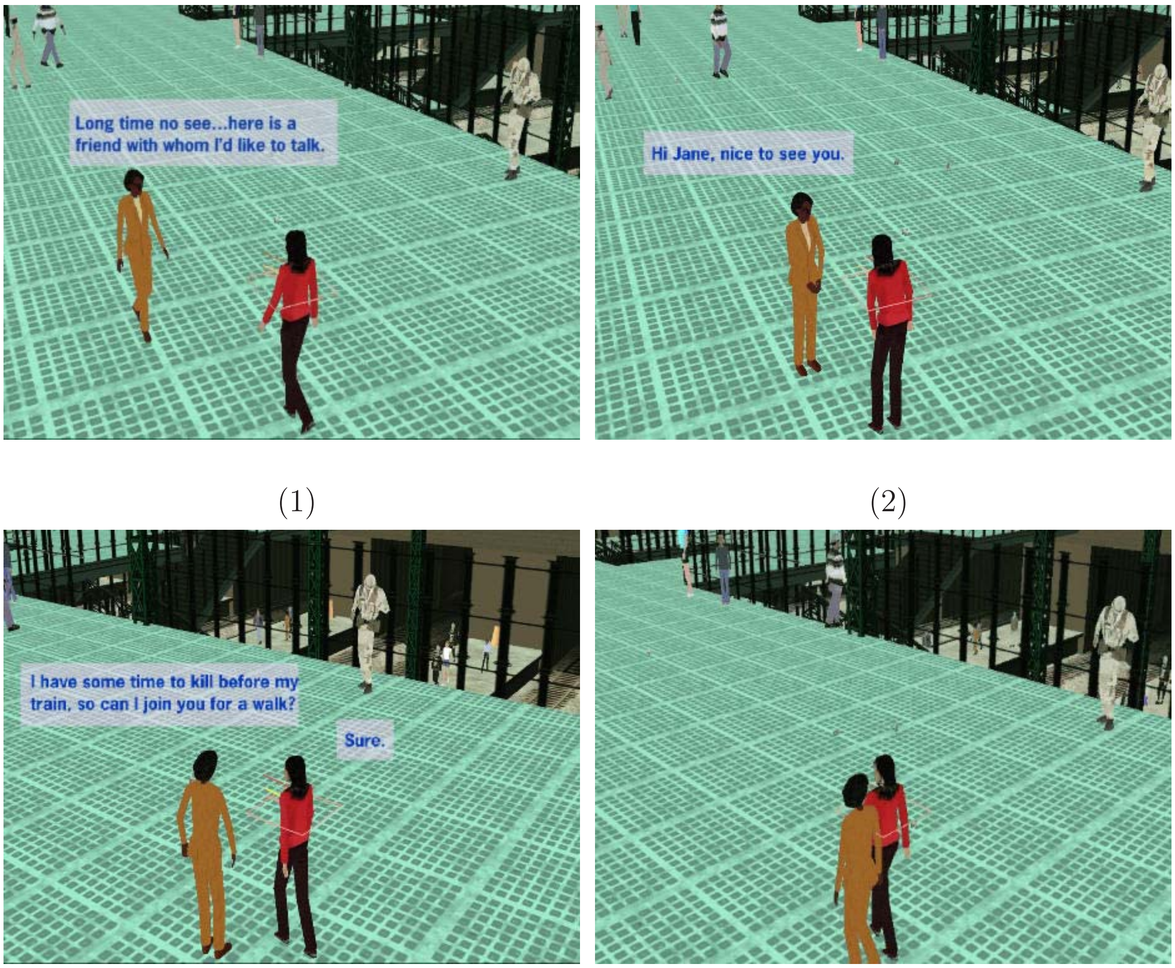


Figure 8.5: Partnering behavior model: rejecting the request.



(1) (2)

Figure 8.6: Partnering behavior model: run into each other.

Chapter 9

Conclusion

9.1 Summary

This thesis has introduced a novel framework for advanced behavioral modeling in virtual humans, which addresses the challenging open problem of simulating complex interactions between people in urban settings. Based on hierarchical decision networks, our framework has aspired to handle uncertainties, model perceptual focus of attention, incorporate personality traits, and facilitate the encoding of behavior rules. Combining probability, decision, and graph theories, it enables us to create autonomous characters that can make nontrivial interpretations and arrive at intelligent decisions dependent on multiple considerations. We demonstrated our framework in behavioral animation scenarios involving interactions between autonomous pedestrians, including an elaborate emergency response animation.

To our knowledge, ours is the first attempt to design a virtual human behavior modeling system and architecture based on decision networks, and the first use of decision networks in computer graphics. Differentiating from traditional crowd simulation, we have concentrated on modeling individual behaviors, which requires that the character be capable of making rational decisions based on the independent assessment of its own environment. This tends to be more complicated and it is hard to achieve using the typical approach of a rule-based system. To achieve this goal, we have developed a decision network framework, which offers several advantages. It can handle uncertainties, which are largely ignored in previous behavior models. Having the characters act based on possibly imprecise interpretation of their surroundings rather than on exact information gained through querying the world database yields automatic animation that more realistically approximates how real people behave, including nontrivial interactions between autonomous characters.

As decision networks offer a compact, informative representation of the world and the behavior system, our framework makes it easier for character designers to encode commonsense knowledge into behavior rules that govern the characters' decisions. It also facilitates modifications, such as adding new parameters to behavior models. Different personality traits and weight settings for relevant factors in the decision making process lead to different behaviors. Observation of the behavior resulting from various personality and internal factor settings can help in validating the decision networks. Our results demonstrate the effectiveness of our framework. Its hierarchical structure keeps the computational complexity at bay, and its clarity provides insights into behavior mechanisms and facilitates the design, implementation, and debugging of behaviors.

9.2 Advantages

Our new framework has several advantages over existing behavioral modeling techniques including scripts and finite state machines, which are mostly rule based and use conditional statements to specify behavior rules.

In order to construct behavior models in general, one has to consider what decisions need to be made, what factors we want to model, how much each of them should influence the final decision, and how they influence each other. This is a necessary exercise regardless of the behavioral modeling approach. Our framework provides a clear, systematic way to represent the fundamental considerations, because as we think about the decision making process, it is natural to chart the relevant causal relationships. Decision networks offer a natural, graphical network structure representation and visualization of these relationships.

Our framework also provides a rigorous and elegant mathematical structure to organize the complicated relationships among the relevant variables in the behavioral model. The weight each input factor carries on the final decision, and the character's preferences can be easily scaled and specified as probability values and utility values. By contrast, to represent the same relationships in terms of conditional statements, one has to explicitly define a rule for each possible relationship. The specification of all the interrelationships among variables using rules is much more tedious and difficult, and the nested conditional statements can be sensitive to the order of the rule specification, as a specific order may create certain bias in the final decision.

Moreover, the reasoning structure behind the rules are much more difficult to visualize and communicate than with a graphical model. Additionally, with decision networks, all the preferences and influencing factors are kept in clearly defined and compact structures. Given the same network topology, we can have characters reasoning and acting quite differently by specifying different probability and utility settings. This cannot be accomplished as easily with rule-based systems, including scripts and finite state machines.

Decision networks have mathematically sound inference algorithms, and provide modular representation of uncertain knowledge. With these capabilities, the characters act based on their interpretations about their environment and about each other, rather than exact information extracted from the direct querying of the database, hence making the simulation more realistic. As explained in Chapter 4, rule-based systems cannot handle uncertainties, and they lack the powerful inferencing capabilities that decision networks have.

As shown in Section 6.1.4, once the networks are constructed, they can automatically generate results for all possible input parameter combinations, including uncertain inputs, which are represented with probabilities. This makes the behavioral model flexible and able to respond sensibly to a wide range of changes in the character's internal and external environment. The network specification and evaluation can be carried out with commercially available software. Specifying rules with conditional statements that can cover the same wide range of potential inputs is difficult and time consuming.

As was discussed in Section 6.1.2, decision networks facilitate parameter fine tuning and debugging. It is easy to identify what contributes to the final decision, and modifications take only minutes to complete, as the structure is so clear, compact and modular. However, debugging with rule-based systems tend to be a painful process. The more complicated the relations that need to be represented, the more error prone and difficult to debug the corresponding rule-based system becomes.

Adding a factor to the networks, may it be a new cue observable or an additional personality trait to be considered, simply requires the addition of a new node plus the corresponding link(s) indicating its relationship with other nodes in the network, as well as its associated probabilities. Examples given in Section 6.1.5 illustrate how easy it is to add emotional components to an existing network. With rule-based systems, extension can easily involve changes to many rules. Identifying all the rules that will be affected by the extension alone can be challenging.

9.3 Applications

Although this thesis is an initial attempt to introduce decision networks to behavioral modeling in the field of computer graphics, we have seen great potential in its application. Computer games are a fast growing industry with more and more emphasis given to gameplay. How to make computer-controlled characters seem more "intelligent" and "engaging" is an important dimension in the success of a game. Our framework's ease of behavior specification and power in making interpretations will facilitate the creation of game characters responsive to their environment and to each other. These characters will be able to make rational decisions based on their uncertain perception of their surroundings and their assessment of the player's intentions. Through varying network parameter settings and/or varying input parameters to the networks, one can quickly create characters that react quite differently even when faced with the same scenario, hence enriching the player's gaming experience.

Computer graphics is playing an increasingly important role in the motion picture industry. Another potential application for our framework is in movie and special effects production. For example, crowd behavior generation is often needed in these applications. Empowered by the ability to simulate personality and emotional effects, our framework can serve to give the characters that are following a crowd an individual touch. While being a member of a large group, the character can still preserve individual differences.

Urban planning is concerned with the design and social environment of municipalities and communities. Populating an urban design with virtual characters to simulate how real humans will use the space can help to test the design and identify potential problems. It is much easier and less expensive to fix the design than to make modifications during or after physical construction. Our framework can be exploited in such applications to take advantage of the ease with which designers can guide the virtual characters following commonsense heuristics about people's general preferences and habits for using such places. Given the decisions that can be made based on a wide range of input parameters, the autonomous characters can be made sensitive to certain aspects of their environments, hence supporting focused testing of required aspects of the urban design.

Virtual human modeling can also assist training. It is often better to give the trainee an interactive and immersive experience to achieve the best results. Our framework facilitates the simulation of interaction among characters, and consequently it potentially offers a good platform for building training scenarios. It can be helpful to analyze and even predict the trainee's behaviors for the instructor to improve a training plan. The decision networks' ability to make both interpretation and prediction makes it a good candidate to fulfill this task.

9.4 Limitations

Using our novel framework, we have developed several behavior models that demonstrated nontrivial interactions between virtual pedestrians. However, real human interaction is of course much more complicated and diverse. Thus, this thesis should be viewed as an initial attempt in applying the decision network framework to computer graphics for human behavior modeling. We will discuss the possible expansion of the behavior repertoire momentarily.

In our probabilistic approach, the full joint probability distribution can answer any possible queries, but the large computation required would make it intractable to handle complex problems.

The compactness of the decision network makes it feasible to handle domains with many variables. However, the decision network is a correct representation of a domain only if each node is conditionally independent of its predecessors in the node ordering, given its parents. For domains in which each variable can be influenced directly by all other variables, the specification for conditional probability tables will be the same as specifying the full joint probability distribution. Hence, the decision network may not be suitable for handling domains with co-related decision factors [Jensen 2001]; i.e., it is difficult to apply decision networks to problems with many non-independent factors.

As the decision network is an acyclic graph, it is not an appropriate solution for problems that must be represented with cycles or that involve the modeling of loops or repetitions. Any behavior that involves cyclic causal chains or recursive plans fall into this category. For example, suppose we want to model a character in a pub. When the character is unhappy, it drinks heavily, but when the character drinks heavily, it feels even more unhappy. So the emotion unhappiness and the drinking behavior form a vicious circle. They are mutually causal, but with an acyclic graph, we cannot represent such a cycle. Given the strength in uncovering information from conditional probabilities, we can try to focus on time series of such problems, and regard the causation as a temporal phenomenon; i.e., assuming one thing causes another by preceding it. By doing so, inferring mutual causation and recursion becomes intractable, but for animation purposes we can still have the characters make reasonable decisions. In this example, we will break them into two separate networks, one has a link from unhappiness to drinking indicating this causal relationship, and the other has a link from drinking to unhappiness. The designer must specify which network should precede the other in the order of evaluation. Given the acyclic nature, the relationships that can be represented at any given time must be either one-way causal influences, or net influences on steady-state conditions over a given time frame [Borsuk et al. 2004]. One way to represent feedback within the system is to employ dynamic decision networks, which supports the modeling of probabilistic relationships for variables within and between time steps.

9.5 Future Work

Our framework offers several opportunities for future research, among them the expansion of the behavioral repertoire, the user interface, improving the motor system, adding a temporal component, and performing sequential decisions, prior and conditional probability settings, and Bayesian parameter learning, each of which we will now discuss in turn.

Expansion of the Behavioral Repertoire: A natural avenue of future work within our framework is to develop additional behavior models. An expanded behavioral repertoire can enrich the complexity and enhance the realism of the interaction between virtual humans. For example, a behavior model can be developed that enables the characters to interact not only pairwise, but also in larger groups. We can expand our interaction models to simulate more sophisticated coordination and cooperation behaviors among multiple characters.

Another direction for expansion is to extend the existing behavior models. Human behaviors involve a complicated decision process, which often has to take many factors into consideration, including external stimuli, internal factors, and personality. Personality is multifaceted. In 1936, Allport and Odbert [1936] extracted 4,500 personality-describing adjectives out of 18,000 personality-describing words that they found in two of the most comprehensive English dictionaries available at the time, which they regarded as describing observable and relatively permanent traits. Many traits may work together to influence a particular decision. In our behavior models, we have tried to simplify the decision process by considering only a few traits that cause prominent effects. Our simulation results demonstrate that our models are capable of producing a variety of different reactions due to individual differences. More elaborate models would necessitate the consideration of additional factors, which would require an expansion of existing behavior models.

Another potential motivation for model expansion is the fact that the limitations of the motor system restricts the available cues that the characters can observe to make interpretations of their surrounding environment. As the motor system improves, more cues may become available and they can easily be incorporated into the models. Hence, the framework should be easily expansible to accommodate this need.

The hierarchical structure of our framework keeps the components of the decision making process modular. Components are kept in separate networks and the connection between them are clearly expressed by how they are linked together in the higher level network. This facilitates communication as it provides a clearer view of how the decision is made. Expansion of the model need only involve the lower level networks associated with the components concerned. Networks corresponding to other components need not be changed. Higher level networks in the hierarchical structure need only be updated when it has been necessary to add lower level networks corresponding to new components that must be considered in the decision process.

Adding a User Interface: In our current system, the parameters associated with the virtual characters, including their characteristics, are specified in a configuration file, which is loaded before the simulation begins. An intuitive user interface should be developed to facilitate the graphical construction of behavior models. The user should also be able to input personality parameters and internal factors on the fly, and their values will be entered as evidence within the behavior models such that the simulation will immediately reflect the modifications. This sort of interactivity will enable animators to more easily detect potential problems with the behavior model by making adjustments on the fly and observing the resulting behavior. It can also potentially support personality studies, as users could try different personality settings more easily and observe their effects on human behaviors.

Improving the Motor System: Though our framework is general in nature, the limitations of our current DI-Guy-based motor system constrains the observable cues that are available in making interpretations; hence, the quality of the animations that can be generated. With a more advanced motor system that is capable of generating more detailed human movements, including facial expressions, and affords more precise control over motor movements, our decision network framework can access more observable cues in making interpretations over the character's environment, especially its assessment of other characters' intentions. This will in turn enhance the character's decision making.

Adding a Temporal Component: In our current framework, we assume that the decision at any time step depends only on observations made at that time step or at most in the previous time step. Some complex decisions, however, may depend on a substantial history of previous observations. Moreover, in the current system we concentrate on making episodic decisions; i.e., decisions made in one shot, where the utility for the outcome of each action is well defined. For sequential decisions, the utilities actually depend on a sequence of decisions. To model such decisions, it will be necessary to augment our model with a temporal component. A possible extension to our current framework, which should preserve its ability to handle partially observable, uncertain environments and unexpected observations, is to use dynamic decision networks [1990].

Training the Network from Data: The prior and conditional probability distributions are currently set by designers subject to commonsense considerations. While facilitating the behavior model specification, they may be more arbitrary than desirable. As data become available from human behavior studies, they can potentially be used to enhance the probability distribution settings for the models. Techniques have been developed in training Bayesian networks from data. They can be applied to enhance the quality of the probability distribution settings, which will in turn lead to more accurate behavior models.

Bayesian Parameter Learning: Intensive research has been done on learning Bayesian networks. These techniques can potentially be applied to improve the probability distributions further. For example, the accuracy of the interpretations needed to arrive at a decision depends on the associated network structure and its probability distributions. By querying the system database directly, we can determine the exact answers to what needs to be interpreted. These exact answers can then be compared with the interpretation results to fine tune the associated probability distributions and even the decision network structures if necessary. The improvements made in the networks will in turn improve the quality of behaviors being simulated.

In closing, this thesis has demonstrated for the first time that decision networks can be a powerful framework for developing behavior modeling systems for autonomous virtual humans. Given the further development of our framework in the above suggested directions, one should be able to populate virtual worlds with human characters capable of making complex, perceptually-motivated decisions and exhibiting realistic mutual interactions, while at the same time being amenable to the input of an animator who wishes to direct their behaviors. These smart characters will also be capable of learning from their experiences, thereby improving their decision-making skills.

References

Aji, S., and McEliece, R. 2000. The generalized distributive law. In *IEEE Transactions on Information Theory*, vol. 46, 325–344.

Allport, G. W., and Odbert. 1936. Trait names: A psycho-lexical study. *Psychological Monographs 47*, 211.

American Psychiatric Association. 1994. *Diagnostic and Statistical Manual of Mental Disorders*, 4th ed. Author, Washington, DC.

Bacchus, F., and Grove, A. 1996. Utility independence in qualitative decision theory. In *KR'96: Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Francisco, California, L. C. Aiello, J. Doyle, and S. Shapiro, Eds., 542–552.

Bacchus, F., Dalmao, S., and Pitassi, T. 2003. Algorithms and complexity results for #sat and bayesian inference. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington, DC, 340.

Badler, N. I., Allbeck, J. M., Zhao, L., and Byun, M. 2002. Representing and parameterizing agent behaviors. In *Computer Animation*, 133–143.

Ball, G., and Breese, J. 2000. Emotion and personality in a conversational character. In *Embodied Conversational Agents*, MIT Press, Cambridge, MA, 189–219.

BARON-COHEN, S., RIVIERE, A., FUKUSHIMA, M., FRENCH, D., HADWIN, J., CROSS, P., BRYANT, C., AND SOTILLO, M. 1996. Reading the mind in the face: A cross-cultural and developmental study. *Visual Cognition 3*, 39–59.

BARON-COHEN, S. 1995. *Mindblindness: An Essay on Autism and Theory of Mind.* MIT Press, Cambridge, MA.

BATES, J., LOYALL, A. B., AND REILLY, W. S. 1994. An architecture for action, emotion, and social behavior. In *MAAMAW '92: Selected papers from the 4th European Workshop on on Modelling Autonomous Agents in a Multi-Agent World, Artificial Social Systems*, Springer-Verlag, London, UK, 55–68.

BISHOP, C. M. 1995. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, UK.

BLAKEMORE, S., AND DECETY, J. 2001. From the perception of action to the understanding of intention. *Nature Reviews Neuroscience 2*, 8, 561–567.

BLUMBERG, B. M., AND GALYEAN, T. A. 1995. Multi-level direction of autonomous creatures for real-time virtual environments. *Computer Graphics 29*, Annual Conference Series, 47–54.

BLUMBERG, B., DOWNIE, M., IVANOV, Y., BERLIN, M., JOHNSON, M. P., AND TOMLINSON, B. 2002. Integrated learning for interactive synthetic characters. In *Proc. ACM SIGGRAPH 02*, 417–426.

BORSUK, M., STOW, C. A., AND RECKHOW, K. 2004. A bayesian network of eutrophication models for synthesis, prediction and uncertainty analysis. *Ecological Modelling*, 173, 219–239.

BREAZEAL, C. 2003. Emotion and sociable humanoid robots. *Int. J. Hum.-Comput. Stud. 59*, 1-2, 119–155.

BROOKS, R. A., BREAZEAL, C., MARJANOVIC, M., SCASSELLATI, B., AND WILLIAMSON, M. 1999. The cog project: Building a humanoid robot. In *Computation for Metaphors*, Lecture Notes in Artificial Intelligence 1562. New York, Springer, Analogy and A. C. Nehaniv, Eds., 52–87.

CASSELL, J., BICKMORE, T., CAMPBELL, L., CHANG, K., VILHJÁLMSSON, H., AND YAN, H. 1999. Requirements for an architecture for embodied conversational characters. In *Proceedings of Computer Animation and Simulation '99*, Springer Verlag, Berlin, 109–120.

CHOPRA-KHULLAR, S., AND BADLER, N. 2001. Where to look? automating attending behaviors of virtual human characters. *Autonomous Agents and Multi-agent Systems 4*, 1/2, 9–23.

CONDORCET. 1793. Plan de constitution, presente a la convention nationale les 15 et 16 fevrier 1793. *Oeuvres 12*, 333–415.

COWELL, R. G., LAURITZEN, S. L., DAVID, A. P., SPIEGELHALTER, D. J., AND SPIEGELHATER, D. J. 1999. *Probabilistic Networks and Expert Systems*. Springer-Verlag New York, Inc., Secaucus, NJ.

DE FINETTI, B. 1937. La prevision: Ses lois logiques, ses sources subjectives, annales de l'institut henri poincare 7, 1-68. translated into english by henry e. kyburg jr., "foresight: Its logical laws, its subjective sources". In *Studies in Subjective Probability*, Wiley, New York, H. E. K. Jr. and E. Howard E. Smokler (1964, Eds., 53–118.

DEVILLERS, F., DONIKIAN, S., LAMARCHE, F., AND TAILLE, J. 2002. A programming environment for behavioural animation. *Journal of Visualization and Computer Animation 13*, 263–274.

DEWEY, J. 1910. How we think. *Middle Works 6*, 177–356.

Doyle, J., and Thomason, R. H. 1999. Background to qualitative decision theory. *AI Magazine 20*, 2, 55–68.

Doyle, J. 1980. A model for deliberation, action and introspection. Technical Report 419, MIT.

Doyle, P. 1999. Virtual intelligence from artificial reality: Building stupid agents in smart environments. In *In Working Notes of the 1999 AAAI Spring Symposium on Artificial Intelli-gence and Computer Games*, AAAI Press.

Egeth, H. E., and Yantis, S. 1997. Visual attention: Control, representation, and time course. *Annu. Rev. Psychology 4*, 269–297.

Egges, A., Zhang, X., Kshirsagar, S., and Thalmann, N. M. 2003. Emotional communication with virtual humans. In *Multimedia Modelling*.

Fausett, L. 1994. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ.

Funge, J., Tu, X., and Terzopoulos, D. 1999. Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. In *Proc. ACM SIGGRAPH 99*, 29–38.

Goldberg, L. R. 1981. Language and individual differences: The search for universals in personality lexicons. *Review of personality and social psychology*, 141–165.

Guye-Vuilleme, A., and Thalmann, D. 2001. A high-level architecture for believable social agents. *Virtual Reality Journal 5*, 95–106.

Howard, R., and Matheson, J. 1981. Influence diagrams. *In Howard, R. and Matheson, J., editors, Readings on the Principles and Applications of Decision Analysis II*, 721–762.

Huang, C., and Darwiche, A. 1994. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning 15*, 3, 225–263.

Hy, R. L., Arrigoni, A., Bessière, P., and Lebeltel, O. 2004. Teaching bayesian behaviours to video game characters. *Robotics and Autonomous Systems 47*, 3, 177–185.

Jaakkola, T., and Jordan, M. I. 1999. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research 10*, 291–322.

Jensen, F. V. 2001. Bayesian networks and decision graphs. In *Bayesian networks and decision graphs*. Springer.

Johnson, M. B. 1995. *Wavesworld: a testbed for constructing three-dimensional semi-autonomous animated characters*. PhD thesis, Massachusetts Institute of Technology.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. 1998. An introduction to variational methods for graphical models. In *Learning in Graphical Models*, M. I. Jordan, Ed. Kluwer Academic Publishers, Boston.

Korb, K. B., and Nicholson, A. E. 2004. *Bayesian Artificial Intelligence*. Chapman & Hall, Boca Raton.

The authors develop an approach to artificial intelligence based on Bayesian networks with a full treatment of learning, knowledge engineering, and elicitation. Their web site at http://www.csse.monash.edu.au/bai/ offers valuable supplementary material.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory 47*, 2, 498–519.

Kshirsagar, S. 2002. A multilayer personality model. In *Proceedings of the 2nd interna-tional symposium on Smart graphics*, 107–115.

Lebeltel, O., Bessiere, P., Diard, J., and Mazer, E. 2004. Bayesian robot programming. *Autonomous Robots 16*, 49–79.

Maes, P. 1989. How To Do the Right Thing. *Connection Science 1*, 3, 291–323.

Matarić, M. J. 1997. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence 9*, 2-3, 323–336.

McCulloch, W., and Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys. 5*, 115–133.

Mintzberg, H., Raisinghani, D., and Theoret, A. 1976. The structure of 'unstructured' decision processes. *Administrative Sciences Quarterly 21*, 246–275.

Musse, S. R., and Thalmann, D. 1997. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Computer Animation and Simulation '97, Proc. EG Workshop*, Springer-Verlag, Budapest, 39–52.

Noser, H., and Thalmann, D. 1996. The animation of autonomous actors based on production rules. In *Proc. Computer Animation '96*, IEEE Comp. Soc. Press, 47–57.

Oliver, N. M., Rosario, B., and Pentland, A. P. 2000. A bayesian computer vision system for modelling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (August).

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, CA.

Perlin, K., and Goldberg, A. 1996. Improv: a system for scripting interactive actors in virtual worlds. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM Press, 205–216.

Peters, C., and Sullivan, C. O. 2003. Bottom-up visual attention for virtual human animation. In *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, IEEE Computer Society, Washington, DC, 111.

Peters, C. 2005. Foundations for an agent theory of mind model for conversation initiation in virtual environments. In *Proceedings of the AISB '05 symposium on Virtual Social Agents: Mind-Minding Agents*.

Ramsey, F. P. 1931. Truth and probability. In *The Foundations of Mathematics and Other Logical Essays*, R. B. Braithwaite, Ed.

Reilly, W. S., and Bates, J. 1995. Natural negotiation for believable agents. Technical Report CMU-CS-95-164, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA, June.

Reynolds, C. 1987. Flocks, herds, and schools: A distributed behavioral model. *Proceedings of ACM Computer Graphics 21*, 4 (Jul), 25–33.

Roth, D. 1993. On the hardness of approximate reasoning. In *Proc. of the 13th IJCAI*, 613–618.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence - A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ.

Savage, L. J. 1954. *The Foundations of Statistics*. Wiley, New York.

Shachter, R. D. 1990. Evaluating influence diagrams. In *Readings in Uncertain Reasoning*, G. Shafer and J. Pearl, Eds. Kaufmann, San Mateo, CA, 79–90.

Shao, W., and Terzopoulos, D. 2005. Autonomous pedestrians. In *Proc. SIGGRAPH/EG Symposium on Computer Animation (SCA'05)*, 19–28.

Shawver, D. M. 1997. Virtual actors and avatars in a flexible user-determined-scenario environment. In *VRAIS '97: Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)*, IEEE Computer Society, Washington, DC, USA, 170.

Silverman, B. G., Johns, M., Cornwell, J., and O'Brien, K. 2006. Human behavior models for agents in simulators and games: Part i: Enabling science with PMFserv. *Presence: Teleoper. Virtual Environ. 15*, 2, 139–162.

Simon, H. 1960. *The New Science of Management Decision.*

Tatman, J. A., and Shachter, R. D. 1990. Dynamic programming and influence diagrams. *IEEE Transactions on Systems 20*, 2, 365–379.

Tu, X., and Terzopoulos, D. 1994. Artificial fishes: Physics, locomotion, perception, behavior. *Computer Graphics (Proc. SIGGRAPH'94)*, 43–50.

Ulicny, B., and Thalmann, D. 2001. Crowd simulation for interactive virtual environments and vr training systems. In *Proc. Eurographics Workshop on Animation and Simulation (EGCAS 01)*, Springer-Verlag, New York, 163–170.

von Neumann, J., and Morgenstern, O. 1944. *Theory of Games and Economic Behavior (first edition).* Princeton University Press.

Zadeh, L. A. 1988. Fuzzy logic. *Computer 21*, 4 (April), 83–93.