An Emergent Framework For Realistic Psychosocial Behaviour In Non Player Characters

Christine Bailey
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada
+1-519-661-2111
cdbailey@csd.uwo.ca

Michael Katchabaw
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada
+1-519-661-4059
katchab@csd.uwo.ca

Abstract

This paper introduces a framework for emergent psychosocial behaviour in non player characters in video games. This framework uses concepts behind emergent gameplay to support the mechanics of designer-defined psychological and social concepts, undefined circumstances, and emergence. Based on this framework, a prototype system has been developed. This prototype has been evaluated for realistic emergent behaviour, and has been shown through experimentation to succeed in supporting emergent psychosocial behaviour. The work to date on the framework is encouraging and quite promising for continued work in this area in the future.

Author Keywords

Realistic behaviour, psychosocial behaviour, non player character behaviour, emergent gameplay, stimulus-response systems.

Introduction

The field of artificial intelligence in games is a broad, yet demanding area of study. In (Russel and Norvig, 2003), artificial intelligence is defined as being concerned with thought processes, reasoning, and behaviour as it applies to human performance as well as ideal intelligence, or rationality. Artificial intelligence in games differs from traditional artificial intelligence in that it is optimized not for the simulation of human performance or rationality, but for entertainment and increased immersion in the game world (Reynolds, 2004; Tozour, 2002).

One of the more active areas of research in game artificial intelligence is making more believable characters, also known as Non Player Characters (NPCs) (Byl, 2004; Funge, 2004; Game Informer, 2006a; Game Informer, 2006b; Gruenwoldt et al, 2007; Guye-Vuilleme and Thalmann, 2001; Lawson, 2003; Prendinger and Ishizuka, 2001; Rizzo et al, 1997). This has been investigated through adding simple mental states to characters (Byl, 2004; Funge, 2004) and more complex mental states such as emotions (Funge, 2004; Rizzo et al, 1997). NPCs can also be implemented to have memory (Funge, 2004), and to be able to forget events (Alt and King, 2002; Grond and Hanson, 1998). More complex believability has been achieved through the creation of reputation systems where NPCs share opinions of the game player based on their behaviour in the game world (Alt and King, 2002; Brockington, 2003; Electronic Arts, 2005; Grond and Hanson, 1998; Gruenwoldt et al, 2007; Russel, 2006), and even implementation of social awareness, including the maintenance of interaction histories between NPCs (Tomlinson and Blumberg, 2002), changes in attitudes and familiarity between NPCs (Prendinger and Ishizuka, 2002), changes and propagation of opinions (Cole, 2006), and the design of architectures to enable social identities (Guye-Vuilleme and Thalmann, 2001). Unfortunately, there have been few, if any, attempts to unify psychosocial behaviour in NPCs so as to include emotions, personality, and individual social relationships. Consequently, the focus of this research has been to design a unified psychosocial framework for characters in real-time interactive entertainment simulations, such as video games.

There has been much support for the creation of believable and social characters (Byl, 2004; Game Informer, 2006a; Game Informer, 2006b; Lawson, 2003; Prendinger and Ishizuka, 2001; Rizzo et al, 1997). Some of the reasons for having believable social characters includes that they support the suspension of disbelief required to immerse a player in the game (Byl, 2004), promote realism (Berger, 2002), create more dramatically interesting game situations (Prendinger and Ishizuka, 2001), as well as supporting game designers in providing a compelling game experience, supporting interactivity, player choice, and replayability (Reynolds, 2004). Some games have already begun implementing social systems in games, including *Bully* (Game Informer, 2006b) and *Assassin's Creed* (Game Informer, 2006a). Unfortunately, the developers have not revealed how they designed and implemented these systems.

One interesting solution to the problem of developing a framework for realistic psychosocial behaviour is to use the concepts behind emergence and emergent gameplay to create and maintain a unified psychosocial system. The concept behind emergence is that simple component-level behaviours often result in complex surprising system-level behaviour (Holland, 1998; Johnson, 2001; Sweetser, 2008; Wooton, 2006). Emergence, or self-organization, has been studied in disciplines as varied as philosophy, biology, chemistry, physics, ecology, neurology, psychology, and computer science, to name but a few (Holand, 1998; Johnson, 2001). Emergent gameplay focuses on implementing a bottom up view of the game world, in that every (static or non-static) object in the world maintains its own needs and reactions to situations that may arise (Wooton, 2006). In doing so, emergent gameplay has the potential to create "a large amount of gameplay experiences from a much smaller set of interconnected game rules", which can lead to a better overall player experience (Pfeifer, 2004).

By having simple component-level mechanics that interact with each other, overall system behaviour is complex, robust, reactive, and unpredictable. Other places this concept has been successfully used are in flocking algorithms (Cole, 2006; Wooton, 2006) and board games (Holland, 1998; Wooton, 2006). Emergent gameplay has been used in computer games, just as it has been used in board games. Games like *SimCity*, *The Sims* (Wooton, 2006), *Deus Ex*, and *Thief: Deadly Shadows* (Smith and Smith, 2004; Ion Storm, 2004) have all used some sort of emergent gameplay, though typically in the form of very simple behaviours. For example, in *Thief: Deadly Shadows,* emergence is used to define how elements like fire, oil, noise, and guards interact (Smith and Smith, 2004). With few exceptions, unfortunately, most developers do not publish how they have (or have not) used emergent gameplay.

This research presents a unified psychosocial framework, and in doing so uses the concepts developed in emergent gameplay to create and maintain realistic and unpredictable psychosocial behaviour for NPCs in video games. The term realistic is applied here to behaviour that is believable, or convincing, to human observers. Emergence may be used to create this type of behaviour by defining emotional and social concepts as components which react to (or are) stimuli, just as objects such as flame were designed as reactive components in games like *Thief: Deadly Shadows* and *Deus Ex*. This framework supports NPCs with emergent psychology and social behaviour, which allows reactions to undefined circumstances and NPC autonomy.

The remainder of this paper is structured as follows. We begin by presenting a discussion of related work in this area. We then introduce the design of our psychosocial framework based on the principles of emergent gameplay. Based on this, we then describe the prototype system implemented as proof of concept of this framework and then discuss our experiences to date in using this prototype system. Finally, we conclude this paper with a summary and discussion of possible directions for future work in this area.

Related Work

The literature in this area is rich with work taking various approaches to exploring aspects of psychosocial behaviour for NPCs (autonomy, emotion, personality, and realistic social behaviour), although none, unfortunately, have integrated all of the components necessary for realistic and believable psychosocial behaviour. This section provides an overview of this related work; a more complete discussion can be found in (Bailey, 2007).

NPC behaviour is often handled using finite state machines to represent state of mind (Byl, 2004; Funge, 2004), or scripting to hard-code behaviour in each possible game situation (Berger, 2002). The main goal for game artificial intelligence is to support the designers in providing a compelling game experience, supporting interactivity, player choice, and replayability (Reynolds, 2004).

One of the factors in creating a compelling experience is suspending the player's disbelief. The AIP (Autonomy Interaction Presence) Cube model (as cited by (Byl, 2004)) states that the requirements of suspension of disbelief are Autonomy of the NPCs, Interaction between the NPCs and the player, and the NPC's Presence in the game world. The work in (Byl, 2004) also implicates personality, emotion, self-motivation, the illusion of life (which includes goals, plans, reactions to the environment, human-like restrictions, and many other behaviours which create the appearance of life), change, and social relationships as being important to character believability.

Some relevant readings in the area of affective computing and emotion synthesis include Funge's definition of emotion as the sum of past events (Funge, 2004), the discussion in (Byl, 2004) of character emotion in the context of games, and Picard's discussion of shifting emotions (Picard, 1995), and Picard in (Picard, 1995) and the overview in (Byl, 2004) of the field of affective computing. Personality in agents is defined as the agent's pattern of behaviours or interactions with its environment (Lawson, 2003). Crawford (Crawford, 2005) outlines one possible personality model for NPCs in games that include intrinsic variables (i.e. integrity, virtue, intelligence, and so on), mood variables, relationship variables (beliefs about another's intrinsic variables), and the readiness to change the previous two variables. Isbister discusses the social psychology research and discusses the traits of agreeableness and dominance and how they can be used to form many different personalities (Isbister, 2006).

Reputation systems – such as those in *Fable* (Alt and King, 2002), *Thief: Deadly Shadows* (Ion Storm, 2004), and *Ultima Online* (Grond and Hanson, 1998) – refer to systems that typically manage NPC opinions of the player (Alt and King, 2002; Electronic Arts, 2005), which are formed immediately and globally among all NPCs upon certain player actions (Alt and King, 2002; Brockington, 2003; Electronic Arts, 2005; Grond and Hanson, 1998). Reputation systems do not maintain individual opinions (Alt and King, 2002; Electronic Arts, 2005), nor opinions about other NPCs, though they may maintain group opinions (Alt and King, 2002; Grond and Hanson, 1998; Ion Storm, 2004).

Some social science concepts of interest in video game research include an agent's roles (Guye-Vuilleme and Thalmann, 2001; Isbister, 2006), cultures and subcultures (Isbister, 2006), norms, values, worldview (Guye-Vuilleme and Thalmann, 2001), and goals (Isbister, 2006). Some papers have attempted to address these (Guye-Vuilleme and Thalmann, 2001; Isbister, 2006). The area of social agents, or Socially Intelligent Agents (SIAs) (Byl, 2004; Prendinger and Ishizuka, 2002) would appear to have much relevance to this research, however many of these "social" agents do not exhibit realistic social behaviour (Cesta et al, 1996; Guye-Vuilleme and Thalmann, 2001; Hogg and Jennings, 2001; Isbister, 2006; Wooldridge, 1992). Many "social" agents implement only communicative behaviour that is used in a multi-agent problem-solving context to reduce resource usage and increase efficiency. Some relevant research in this area includes Tomlinson and Blumberg's remembered interaction histories between agents (Tomlinson and Blumberg, 2002), Prendinger, et al.'s change of attitudes and familiarity assessment between agents (Prendinger and Ishizuka, 2002), Cole's comparison of opinion flow in a multi-agent system to flocking behaviour (Cole, 2006), and Guye-Vuilleme's high level architecture for social agents (Guye-Vuilleme and Thalmann, 2001).

Finally, a particularly interesting approach to problems of providing unique and immersive experiences lies in emergent gameplay (Pfeifer, 2004; Smith and Smith, 2004; Wooton, 2006), as described earlier. As of yet, emergent gameplay is known to have been used in NPC behaviour only in relatively simple situations and behaviours (LeBlanc, 2000; Pfeifer, 2004; Smith and Smith, 2004; Sweetser, 2008; Wooton, 2006), usually to deal with emergent properties of the game world and the objects contained within it. Emergence, however, has not been used to date in attempting to implement complex psychological states or social relationships. Consequently, our current work explores this exciting possibility.

Framework Design

The fundamental design of our psychosocial character behaviour framework is based on the core ideas behind emergence discussed earlier in this paper. Recall that emergence focuses on a bottom-up view of the game world, where simple component-level behaviours interact to form complex system-level behaviour. This is done by making every object in the world self-centered, goal-directed, and responsible for its own needs and responses. Emergent systems are not scripted, nor are they rule-based (Wooton, 2006). Emergent systems also do not focus on algorithmic behaviours, but rather very simple stimulus responses at the component level (Wooton, 2006). It is important to note that while the system itself is not rule-based, the responses of the components of the system can be (Holland, 1998).

To extend these ideas into a psychosocial context, simple psychosocial objects must be defined that can react to psychosocial stimuli and other psychosocial objects, and be able to maintain their own attributes, needs, and responses to the dynamic game environment. Such psychosocial objects and stimuli can include emotions, personality traits, NPCs, groups, or static objects of import to NPCs (such as possessions). In this way, simple component-level psychosocial behaviours will be defined that can interact in complex and interesting ways.

With this in mind, we first discuss the modeling of NPCs from a psychosocial perspective that is necessary to enable to mechanisms for emergence presented. Following this, we then describe the software design of this framework.

Psychosocial Modeling

When considering the development of emergent psychosocial interactions, we must consider what factors will influence and motivate NPC behaviour, and in particular social behaviour. Such attributes may include emotions, personality traits, values, needs (physiological and otherwise), worldview, and culture. In addition, social ties between social objects must be present.

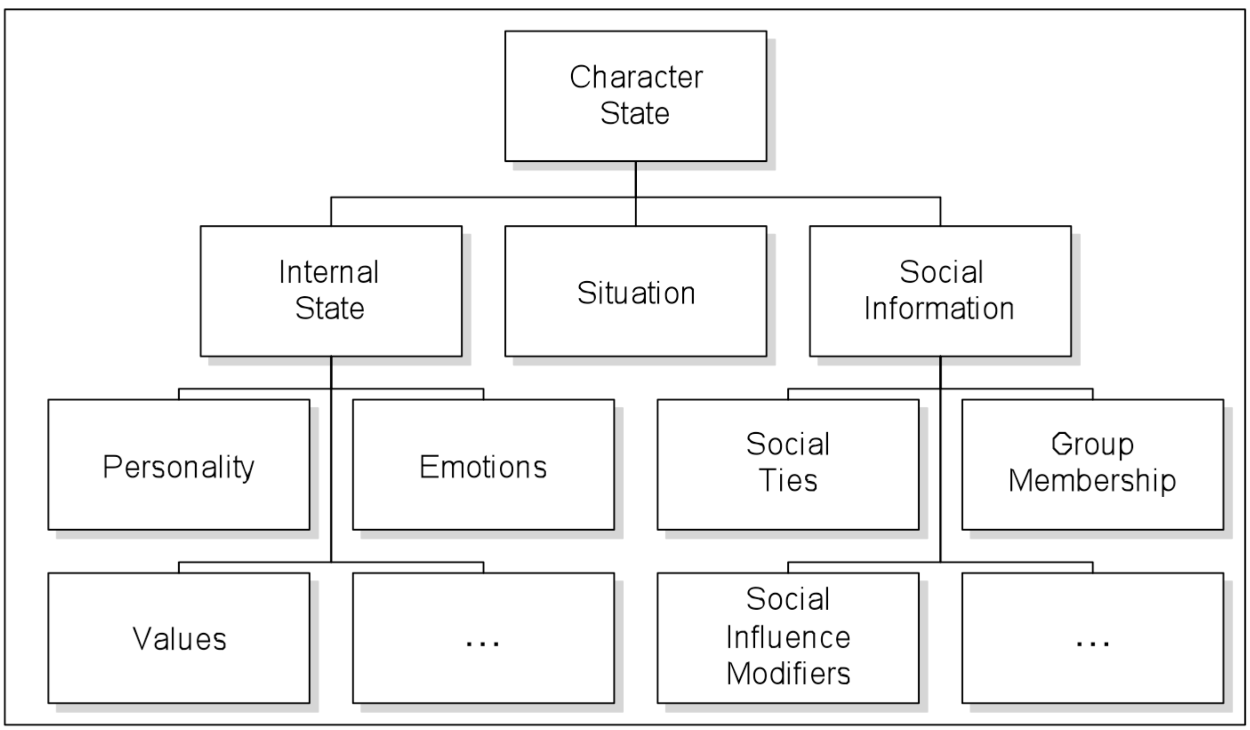

Figure 1: High-Level Character View

Figure 1 depicts a high-level view of a game character. Note the presence of internal states (a personality model, emotional set, values, and so on), social information (social ties, group memberships, and a social influence modifier), and a representation of the current situation (such as the presence of stressors, hostility, and so on).

Not all of the elements discussed here may be necessary to have convincing psychosocial behaviour, nor is this set exhaustive. Attributes may be added or modified to specialize the framework toward the needs of a particular game. However, it is important to note that some form of emotion and personality, at a minimum, should be defined as attributes of an NPC in order to have realistic behaviour, as without personality every NPC would react alike, and without emotion NPCs would likely be too rigid in their reactions to stimuli to appear lifelike.

The following sections will discuss some of the considerations in developing an NPC's internal state and social information.

Internal States

Elements of an NPC psychology that comprise its internal state can include a model of emotion, a personality model, needs, and even values and a worldview (Beaumont, 2008). Since there are different theoretical models of emotions and personality (Funder, 2001), and different game worlds may require different needs, values, worldview and cultural constructs, it should be possible for the game designer to define their own trait models. For example, a designer may wish to use a concise personality model for simplicity, rather than a comprehensive model. To enable this flexibility, these traits should be stored as data, instead of hard-coded into software.

Two examples of possible personality models that could be used are the agreeableness/dominance model of personality (Isbister, 2006) and The Big Five personality model (Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) (Isbister, 2006). A comprehensive emotional set is Ekman's universal emotions (Anger, Fear, Disgust, Surprise, Sadness, and Happiness) (Wortman et al, 2000). Naturally, these models are only samples, and any model that satisfied the game designer's goals could be used instead to define personality and emotion.

Other traits that may warrant consideration are predictability, or consistency (acting as a random modifier to behavioural choices), and sociability (affecting whether the NPC in question wishes appeal to groups that are social, or groups that are dissocial, in nature). Needs, worldview, and values could be defined in the form of general goals and preferences.

Social Information

Social aspects of the environment can be handled by having the NPC maintain social ties to other NPCs, group memberships (what groups the character belongs to), and a social influence modifier that could enable social phenomena such as peer pressure, groupthink, social loafing, or mobs. Social ties can include the strength of the relationship between two social objects and their polarity (whether the relationship is amiable or antagonistic). The social ties can also be modeled to be either symmetric or asymmetric (such that two people may feel different about each other, rather than sharing the same feelings mutually). Group memberships can be represented as a list of groups that the NPC belongs to. A social influence modifier could be defined as a function to modify the influence of internal states. For example, as the number of people around our character increases, our character may change to conform to the group.

The Mechanics of an Emergent NPC Social Psychology

In order to enable emergence in an NPC psychology, it is necessary to understand how emergent systems operate. Emergent systems use simple component-level behaviours that can interact to form complex system-level behaviour (Smith and Smith, 2004; Wooton, 2006). These component-level behaviours operate as stimulus-responses, in that every agent in the world monitors its own perceptions of stimuli and are responsible for responding to those stimuli (Wooton, 2006).

Two example stimulus-response systems are Ion Storm's Stimulus System (Smith and Smith, 2004) (also known as Act/React (Smith, 2007)) and The Sims' object-oriented approach as discussed in (Woodcock, 2000). Ion Storm's Stimulus System was used to enable emergent gameplay by allowing objects to listen for and react to stimuli caused by other objects in the world. Characters in The Sims maintained their own physiological needs and used a stimulus-response system to detect stimulus-broadcasting objects that would meet those needs (McLean-Foreman, 2001; Woodcock, 2000). Note that Ion Storm's Stimulus System dealt only with concrete physical objects, such as candles, oil puddles, and guards while The Sims' approach dealt only with physiological needs.

A Generalized Stimulus-Response System

Central to our approach is a generalized stimulus-response system, shown in Figure 2. This system operates by having objects create and listen for stimuli. By defining a general channel that allows only certain objects to interact with each other through a stimulus that is created and responded to, we have an easy to use mechanism for defining emergent interactions.

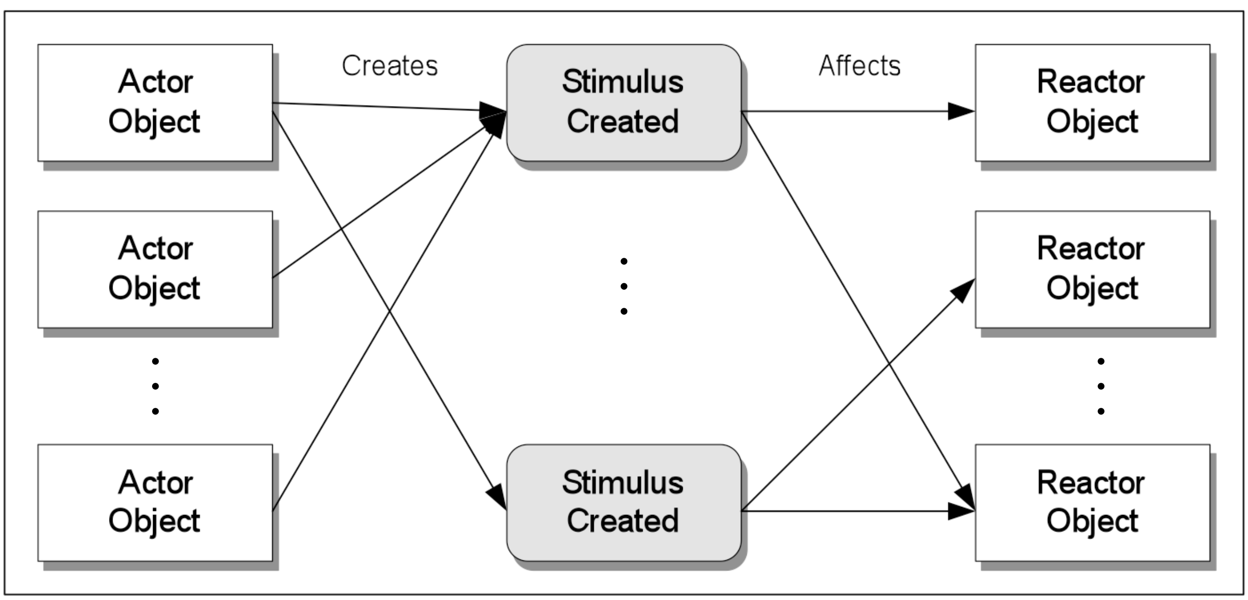

Figure 2: A Generalized Stimulus-Response System

The object or event that begins the interaction by creating a stimulus is referred to as the Actor. We will refer to the object that detects and responds to the stimulus as the Reactor (which may then become an Actor through its response). The stimulus acts as a message that is broadcast from the Actor and is received by the Reactors in the area. Reactors listen for messages with particular stimulus types, since each object will react to different stimuli. For example, in Ion Storm's Stimulus System, an unlit torch would only react to messages with a stimulus type of fire, but not messages with stimulus types of piercing, concussion, or water. In the smart terrain system for The Sims, a Sim who is very hungry will only listen to advertisements of type hunger (McLean-Foreman, 2001), while ignoring other message types.

The stimulus messages can also hold data about the attributes of the associated stimulus. In Ion Storm's Stimulus System, these other attributes included the magnitude of the stimulus, the propagation type, and the fall off radius of the stimulus affect (Smith, 2007). In the smart terrain system for The Sims, these attributes included instructions for how to use the object, relationship data, as well as the fall off radius (Woodcock, 2000). These attributes can be mapped onto different behaviours; it is important to note that Reactors do not have to share the same response to a given stimulus.

3.2.2 A Stimulus-Response System for Psychosocial Behaviour

For an NPC to react in a realistic fashion, they should have the ability to react to the actions of other NPCs, seeing NPCs or objects (i.e. seeing an enemy upon turning a street corner), as well as other events or occurrences in the world, such as natural disasters or accidents. All of these antecedents could be considered the Actors that cause some sort of stimuli. Since these events are to be affected and be reacted to in a psychosocial context, the stimuli created by these events would be psychosocial in nature, such as emotional stimuli (i.e. a witnessed attack may cause an emotional stimulus of fear) or a stimulus that causes a perceived change in situation (i.e. the same witnessed attack may cause the NPC to feel the situation has become hostile).

Psychosocial stimuli are created by Actors as messages and are broadcast to the social objects in the area, as done in Figure 2. These messages hold data about the psychosocial stimulus, such as stimulus source, type, magnitude, propagation, and fall off, as discussed in the example stimulus-response systems earlier in this section. However, these data are modified to reflect the fact that the stimuli are now psychosocial.

Recall that the data in previous stimulus-response systems described physical (or physiological) stimuli and the physical attributes of those stimuli. These attributes included stimulus type (i.e. fire, piercing, and concussion stimuli; hunger, thirst, and fun stimuli), and could also include attributes like magnitude of the affect, the propagation method (how is the stimulus is detected, i.e. fire only causes an affect by contact), and a fall off radius (how does the effect decrease with distance). Since the psychosocial stimulus-response system is not dealing with physical or physiological stimuli, these attributes must be modified somewhat. Stimulus type would refer to emotional and social stimuli instead, such as happiness, sadness, anger, hostile environment, and groupthink. Magnitude of the stimulus would likewise describe the magnitude of the psychosocial effect of the stimulus (i.e. was this a highly emotional event?). Propagation would describe whether the psychosocial stimulus only affects the people directly involved in the event (such as two people greeting each other), if it affects the people witnessing it at a distance as well (such as two people attacking each other). Other types of social stimuli can also be modeled using these stimulus messages, such as gossip and the spread of information. Social stimuli over long distances (such as gossip through a telephone) could be sent as a direct message from one person to another.

It is useful to note that a social fall off radius could serve as a way of simulating social phenomena. By defining the psychosocial stimulus to propagate by radius and defining a fall off rate, an entire group of people can be affected by the behaviour of a few. This may assist in modeling social behaviour such as riots or group-induced panics. While this may not be an accurate model, it can simulate behaviour that appears to be life-like.

The Actors can be anything that broadcasts psychosocial stimuli, including NPCs, objects, and events. The psychosocial stimuli are broadcast to all other social objects in the nearby area. The social objects that are receiving these broadcasts are the interaction Reactors as described in Section 3.2.1. These Reactor objects can be any social object that can react, including NPCs or other social beings, such as groups. Note that non-responsive social objects (such as possessions) will not react to psychosocial stimuli in any way and therefore it would not be desirable to include these objects as Reactors. It may sometimes be desirable to have more general emergent gameplay, by having a stimulus-response system that processes both physical stimuli, and psychosocial stimuli simultaneously (i.e. to allow a game with physical emergent gameplay – such as *Thief: Deadly Shadows* (Ion Storm, 2004) – to have emergent social behaviour as well). In these cases non-socially-reactive objects (like candles) may be treated as Reactors that are only listening for physical, not psychosocial stimuli.

Unlike some of the Reactors described above, social objects, in general, react to, or at least detect, psychosocial stimuli of all types. People tend to react to all emotions, any social situation or stressors they find themselves in, and participate in (or at least detect and make a decision in response to) social phenomena that are occurring around them. In cases where these reactions do not take place, it is because the person (or social object) has gone through some decision-making process to decide how (or how not) to react. This process will require the detection and processing of the stimulus, regardless of its type. Therefore, social objects listen for all types of psychosocial stimuli. However, social objects will typically ignore events that are not relevant to them (i.e. a discussion between two strangers). Therefore social objects can listen for psychosocial stimuli with particular relevance types; some may only be relevant to individual social objects or groups of social objects, while others may be relevant to all.

By defining a psychosocial mechanism in the way described in this section, NPCs and other social objects can react to psychosocial stimuli in realistic as well as unexpected and novel ways to their environment. This can result in chain reactions as reactions to psychosocial stimuli create behaviours or events that can in turn cause more psychosocial stimuli that can be reacted to. This would achieve the goal of having emergent social behaviour in NPCs. By allowing the game player to participate in this mechanism through their actions in the game world, this enables meaningful player decision-making and emergent gameplay.

Each NPC also has a decision-making process to map events and stimuli from the environment to a resultant behaviour. The game designer should have the ability to define how NPCs should react to things in the environment. This should be done in a way that would not necessitate explicitly stating every individual NPC's behaviours separately. To do this, a common set of behaviours should be defined. In order to define this common set of behaviours such that it can be applied over several different NPCs without making specific changes for each NPC, these behaviours can be defined by social role in relation to the NPC in question, instead of particular psychosocial objects. By using social roles to define behaviours, a common set of behaviours can be used by all NPCs to react in the game world.

When a stimulus from the environment is sensed by an NPC, it will refer to a mapping set in data by the game designer that associates the stimulus to a set of appropriate general behaviours. Another designer-defined mapping will associate these behaviours to the traits that influence them. For example, an extraverted NPC will lean towards behaviours involving seeking or interacting with other people, whereas an NPC with high introversion should be less likely to select those behaviours. This mapping can be used in decision-making by taking the set of appropriate behaviours from the stimulus-behaviour set mapping, and choosing from those the most likely behaviour or behaviours to be committed given the NPC's traits and the trait-behaviour mapping.

Software Design

The mechanism explained in the previous section can be implemented by defining a set of behaviours or events that can be reacted to, a set of stimuli that the events create, and a set of social objects that can react to the stimuli. Objects can detect events around them by using a simple event listener, such as the Observer design pattern (Gamma et al, 1995). Observers must subscribe to a subject's event notifications to get these updates. When a subject changes its state, it publishes these changes to all of its subscribers. By using this mechanism, social objects and events in the emergent psychosocial system can notify other social objects of the stimuli they are causing. The observers of those stimuli can then decide whether or not to act on that information. The subjects would be social objects and events (Actors). Observers would be all the social objects in the world that react to the stimuli (Reactors). Observers would subscribe to subject updates when entering within a certain radius, or proximity, of the subject. This area can be defined to be larger to have social objects be more observant of the environment around them, or smaller to conserve the computation resources necessary for this message passing.

Events are passed to the NPCs from the game engine, and actions of the NPCs are sent to the game engine where they would be processed and executed, as shown in Figure 3. Therefore, the NPC logic need not be concerned with how its actions are carried out in the game world.

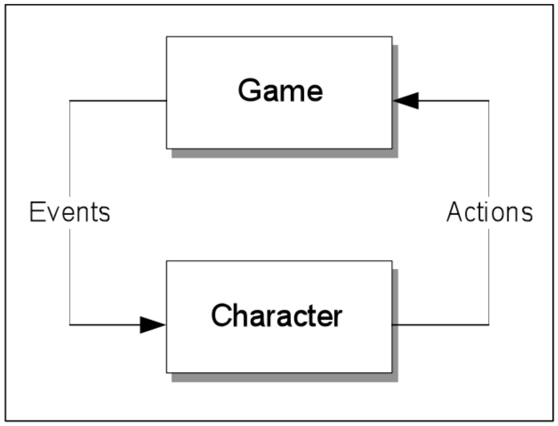

Figure 3: Game Event Flow

Examining the character aspect of Figure 3 in more detail, we have the character architecture shown in Figure 4. In essence, the architecture of a psychosocial NPC perceives the input events from the game as stimuli and outputs resultant behaviour back to the game engine. Some stimuli may be filtered by relevance and thus not cause any resulting output behaviour.

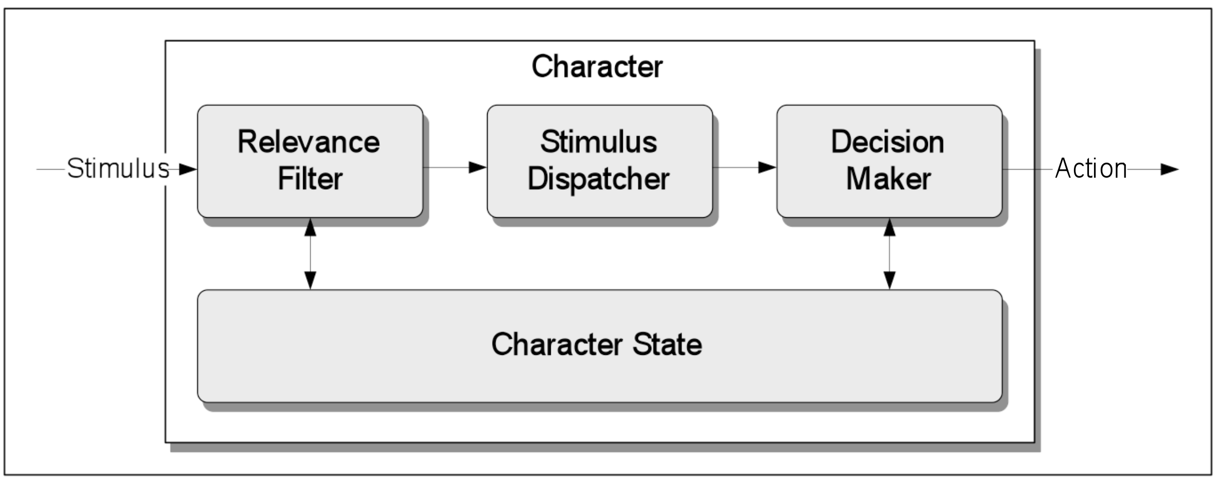

Figure 4: Psychosocial Character Architecture

In order to make an appropriate response, the stimulus information must pass through a Relevance Filter, a Stimulus Dispatcher, and a Decision-Maker. These processes will use information in the Character State as needed, as described below in further detail.

When a stimulus is perceived by the NPC, the Relevance Filter processes it. The Relevance Filter uses the stimulus attributes to determine whether or not the NPC should process this stimulus. The Relevance Filter can have several stages to determine what is relevant, including a group membership filter (to determine if the stimulus is relevant to a group to which the NPC belongs), a propagation method check (to determine if the NPC is within sensing distance of the stimulus), and a magnitude calculation of the stimulus affect (to determine if the stimulus is of significance when it reaches the NPC). In the event that several stimuli are being processed at once, the Relevance Filter can ensure these stimuli are processed in order of priority. The priority given to different messages would be dependent on the state of the NPC. Once the

Relevance Filter determines that a stimulus is in fact relevant to the NPC and what its remaining effect is, the stimulus information and affect is passed on to the Stimulus Dispatcher.

The Stimulus Dispatcher is responsible for determining the type of psychosocial stimulus and the set of possible behavioural reactions and internal state modifications that is appropriate in response to this type of stimulus. The designer would set this information in data as described in Section 3.2.2. This set of possible responses is then passed on to the Decision-Maker.

The Decision-Maker consults the Character State in order to determine whether or not the character should react to the stimulus in question and, if so, how the character should react, given the appropriate set of possible behavioural reactions and internal state modifications in response to the given stimulus type. Any actions taken are passed back to the game engine, and any internal state modifications are made to the Character State.

The Character State refers to the set of attributes that describe the character as a psychological and social being. This was described earlier in Section 3.1, and depicted in Figure 1. For further details of the software design of our emergent psychosocial framework, the reader is urged to consult (Bailey, 2007).

Prototype Implementation

A prototype was made as a proof of concept in the development of this work, implementing the core elements of the emergent psychosocial framework. Microsoft Visual Studio was used to develop the prototype on the Microsoft Windows XP platform. To promote portability of the code, the prototype was developed in C++ using OpenGL to render a simple graphical representation of social objects, the social ties between them, and their moods. A limited character state was implemented including a categorical representation of emotion, a simple personality model, symmetric social ties, and a small behaviour/stimulus set. These elements provide sufficient proof of concept for initial validation and testing. Other aspects of character state such as representations of needs, values, worldview, social influence, and group membership are currently under development.

Experiences To Date

The goal of this work was to develop a unified framework for realistic psychosocial NPC behaviour in video games. To determine if this goal has been met, we must evaluate whether the prototype system described in the previous section allows characters to have realistic psychosocial behaviour. In this section, we discuss a sampling of the social scenarios that were created with the prototype to determine how successfully social and emotional behaviour was realistically modeled.

To determine whether realistic emotional and social behaviour occurred in the prototype, several cases were run with different scenarios to observe any emergent behaviour. The personality traits used were agreeableness and dominance, as described in (Isbister, 2006). The emotions used were anger, sadness, joy, fear, and disgust, based on Ekman's universal emotions (Wortman et al, 2000). NPCs had the ability to detect and react to other NPCs by greeting, hugging, snubbing, or attacking. These actions acted as the psychosocial stimuli. Stimuli were reacted to based on the emotions and personality traits of the reacting NPC with those same actions.

The Angry Individual

The first scenario was based on determining what behaviours would result if a single angry individual was in a group of joyful individuals. To test this, a scenario was set up in which an individual, Brona, was set to be very angry, very dominant and disagreeable, and had a strong dislike for all of the other individuals in the world. The other individuals in the world were all set to have the maximum joy possible, and no other emotions. These individuals had varying personalities and connection strengths between each other, but were, for the most part, on positive terms. This scenario is depicted in Figure 5.

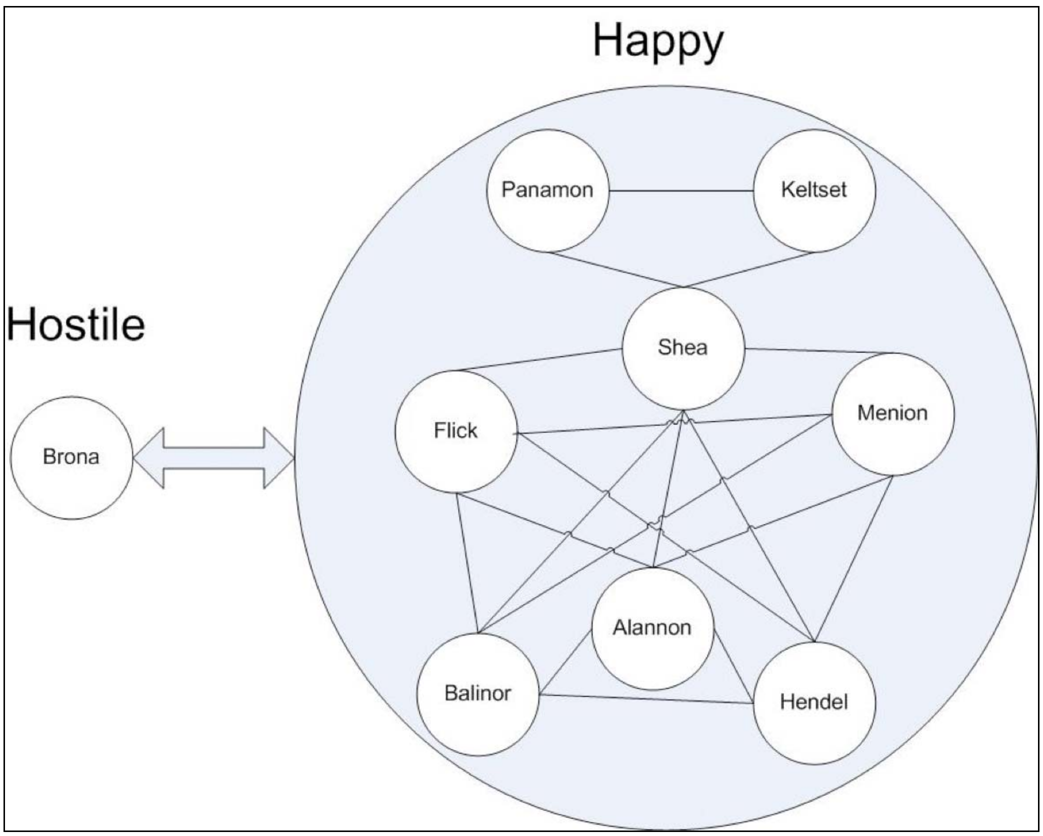

Figure 5: The Angry Individual Scenario

During the run, Brona remained angry regardless of who was around him, and would tend to attack anyone who was around. NPCs who were agreeable and less dominant than Brona (i.e. Shea, Flick, and Hendel) were noted to become sad when around Brona. Individuals who were not agreeable and less dominant than Brona (i.e. Keltset) became fearful when around Brona. Those just as dominant as Brona became angry when around him (i.e. Alannon and Panamon). When two NPCs met who shared a friendly social tie, they would become happy again. Attacks were typically initiated by Brona, however characters with dominance levels comparable to Brona would attack him. Those with lower dominance levels tended to snub Brona in response to his anger.

Figure 6 presents a screenshot of the prototype running this scenario. Red lines denote antagonistic relationships and green lines denote friendly relationships. Individuals are displayed as triangles of the colour denoting their primary emotion (yellow is joy, red is angry, blue is sad, orange is fearful, and green is disgust). Saturation of emotion colours shows how strongly that emotion is felt. Note the emotions of the individuals seen from top to bottom, left to right: Menion is happy, Shea is indifferent, Hendel is indifferent, Brona is angry, Alannon is happy, Flick is sad, Keltset is fearful, Panamon is happy, and Balinor is sad.

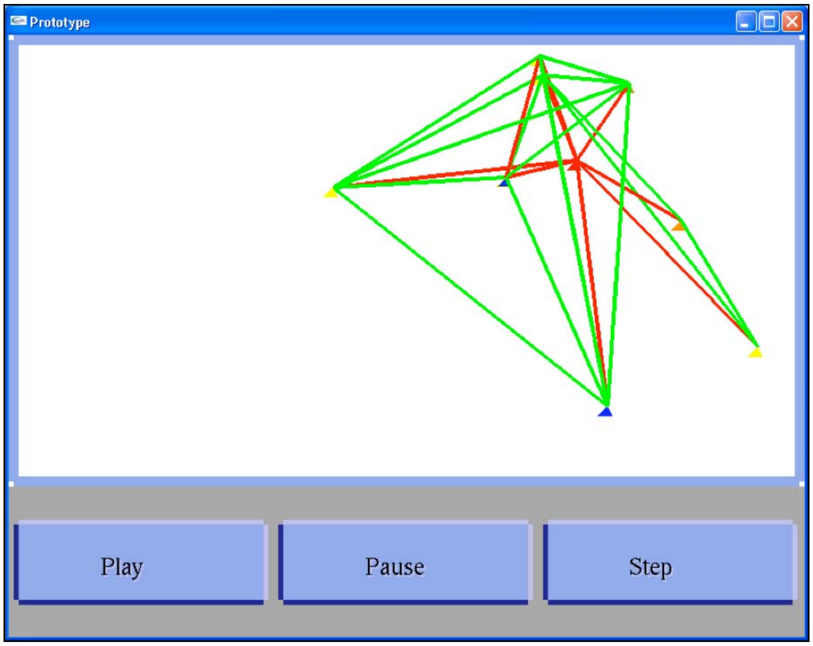

Figure 6: Screenshot of Prototype System During Execution of the Angry Individual Scenario

A Friend in Need, and a Friend in Deed

This scenario was run twice with a slight modification to the emotions. The individuals and social ties were as described in the scenario from Section 5.1, excluding the individual Brona. In the first scenario in this section, all the NPCs had joy set to very high except for Shea, who was set to have very high sadness. The second scenario had only Shea feeling joy, while everyone else was experiencing various random (non-happy) emotions.

In the first case, Shea was quickly cheered when he entered the proximity of any other character. Other characters would greet or hug him depending on their personalities, which would result in an increase in Shea's joy. Every character was then happy, and no change in emotions occurred after this steady state was reached. In the second scenario, joy was gradually spread from Shea through to the other characters. Figure 7 shows how Shea's happiness has begun to spread to the other characters a few moments after the scenario began. Note that joy has spread from Shea, the top-most character in the screenshot to Panamon, Balinor, Hendel, and Flick that are all nearby. The characters furthest from Shea have not yet been affected.

The scenarios executed showed that many unexpected yet appropriate behaviours emerged from the social systems. The observations from these scenarios show that this framework is capable of producing realistic and emergent psychosocial behaviour. A discussion of additional experimental results and experiences can be found in (Bailey, 2007).

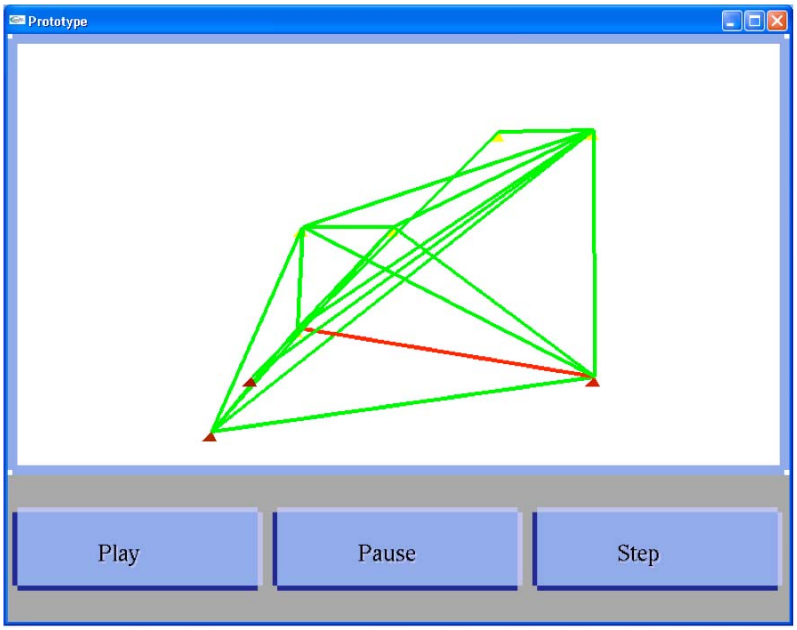

Figure 7: Screenshot of Prototype System Part Way Through the Execution of the Second Variation of the A Friend in Need, and a Friend in Deed Scenario

Concluding Remarks

In conclusion, progress on the framework design and experimental prototype for the emergent psychosocial system has been very positive. Experimentation has shown that the system succeeds quite well in achieving the goals of enabling emergent psychosocial behaviour and interactions, even in the presence of undefined circumstances.

The framework presented in this paper supports designer-defined emotion, personality, and social behaviour. Social concepts such as group memberships, social roles, and values are also supported. The use of emergence has allowed the flexibility to react to undefined circumstances that might be created by the player, while allowing player-authored experience and NPC autonomy.

Future work would include implementing more traits and social concepts, such as social roles, group memberships, social influences and peer pressure, needs, values, and/or representations of individual worldviews. An area of interest is determining whether an accurate simulation could be created with this framework using more accurate models of behaviour. While this emergent psychosocial framework allows character development, it does not handle machine learning, or the assessment of the environmental response to actions. Future work in integrating these fields could be used for more accurate social simulations.

The integration of this system in a fully functional game engine would be another area for future work. This would allow the measurement of performance when used in a graphical real-time interactive game including the processing of other game logic.

An integration with a functional game engine would also allow user studies to determine if the character behaviour and responses are convincing and realistic, and also to determine whether this system results in a better overall gameplay experience. A user study to evaluate usability of the design framework for developers would also be useful in determining the framework's practical value in real world applications.

References

Alt, G. and King, K. (2002). A dynamic reputation system based on event knowledge. In AI Game Programming Wisdom. Charles River Media.

Bailey, C. (2007). Towards Emergent Gameplay: A Framework For Realistic Psychosocial Behaviour In Non Player Characters. MSc. Thesis, Department of Computer Science, University of Western Ontario. December 2007.

Beaumont, L. (2008). Life's Guidance System: Traveling Your Path Through Life. Accessed December 2009 from http://www.emotionalcompetency.com/guidance.htm.

Berger, L. (2002). Scripting: overview and code generation. In AI Game Programming Wisdom. Charles River Media.

Brockington, M. (2003). Building a reputation system: hatred, forgiveness, and surrender in Neverwinter Nights. In Massively Multiplayer Game Development. Charles River Media.

Byl, P.B.D. (2004). Programming believable characters in games. Charles River Media.

Cesta, A., Miceli, M., & Rizzo, P. (1996). Help under risky conditions: robustness of the social attitude and system performance. Proceedings of the First International Conference on Multi-Agent Systems.

Cole, S. (2006). Modeling opinion flow in humans using Boids algorithm & social network analysis. Accessed December 2009 from http://gamasutra.com/features/20060928/cole_01.shtml.

Crawford, C. (2005). Chris Crawford on interactive storytelling. New Riders.

Electronic Arts, Inc. (2005). Reputation, karma and fame. Accessed December 2009 from http://guide.uo.com/combat_5.html.

Funder, D.C. (2001). The Personality Puzzle. W. W. Norton & Company.

Funge, J.D. (2004). Artificial intelligence for computer games. A K Peters.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

Game Informer. (2006a). Assassin's Creed. June 2006, Issue 158. Vol XVI: No 6.

Game Informer. (2006b). Bully. September 2006, Issue 161, Vol XVI: No 9.

Grond, G., and Hanson, B. (1998). Reputation system FAQ. Accessed December 2009 from http://update.uo.com/repfaq.

Gruenwoldt, L., Katchabaw, M., and Danton, S. (2007). Achieving realistic reactions in modern video games. In Worlds in Play. Peter Lang Press.

Guye-Vuilleme, A., & Thalmann, D. (2001). A high-level architecture for believable social agents. VR Journal, 5.

Hogg, L.M.J., & Jennings, N.R. (2001). Socially intelligent reasoning for autonomous agents. IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans, 31(5).

Holland, J. H. (1998). Emergence: From chaos to order. Addison-Wesley.

Isbister, K. (2006). Better game characters by design. Morgan Kauffman.

Jennings, N.R., Sycaro, K., & Wooldridge, M. (1998). A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems, 1.

Johnson, S. (2001). Emergence: The connected lives of ants, brains, cities, and software. Scribner.

Lawson, G. (2003). Stop relying on cognitive science in game design - use social science. Accessed December 2009 from http://www.gamasutra.com/php-bin/letter_display.php?letter_id=647.

LeBlanc, M. (2000). Formal design tools: Emergent complexity, emergent narrative. Proceedings of the 2000 Game Developers Conference.

McLean-Foreman, J. (2001). An interview with Will Wright. Accessed August 2008 from http://gamasutra.com/features/20010501/wright_01.htm.

Pfeifer, B. (2004). Creating emergent gameplay with autonomous agents. Proceedings of the Game AI Workshop at AAAI-04.

Picard, R.W. (1995). Affective computing. Media Laboratory, Perceptual Computing TR 321, MIT Media Lab.

Prendinger, H., & Ishizuka, M. (2001). Social role awareness in animated agents. International Conference on Autonomous Agents.

Prendinger, H., & Ishizuka, M. (2002). Evolving social relationships with animate characters. Proceedings of the AISB-02 Symposium on Animating Expressive Characters for Social Interactions.

Reynolds, B. (2004). How AI enables designers. Accessed December 2009 from http://gamasutra.com/php-bin/news_index.php?story=11577.

Rizzo, P., Veloso, M., Miceli, M., & Cesta, A. (1997). Personality-driven social behavior in believable agents. Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents.

Russel, A. (2006). Opinion systems. AI game programming wisdom 3. Charles River Media.

Russell, S., and Norvig, P. (2003). Artificial intelligence: A modern approach. Prentice Hall.

Smith, H., and Smith, R. (2004). Practical techniques for implementing emergent gameplay: would the real emergent gameplay please stand up? Proceedings of the 2004 Game Developers Conference.

Smith, R. (2007). "RE: [Fwd: Emergent Gameplay presentation]". Email to Christine Bailey. Date emailed: 3 Aug, 2007.

Sweetser, P. (2008). Emergence in games. Charles River Media.

Ion Storm. (2004). Thief: Deadly Shadows. Eidos Interactive.

Tomlinson, B., & Blumberg, B. (2002). Using emotional memories to form synthetic social relationships. Accessed December 2009 from http://citeseer.ist.psu.edu/537507.html.

Tozour, P. (2002). The evolution of game AI. In AI game programming wisdom Charles River Media.

Woodcock, S. (2000). Game AI: The state of the industry. Accessed August 2008 from http://gamasutra.com/features/20001101/woodcock_01.htm.

Wooldridge, M.J. (1992). The logical modeling of computational multiagent systems. PhD Thesis, UMIST, Manchester, October 1992.

Wooton, B. (2006). Designing for emergence. AI game programming wisdom 3. Charles River Media.

Wortman, C.B., Loftus, E.F., Weaver, C., & Atkinson, M.L. (2000). Psychology. McGraw-Hill Ryerson.