Chapter 2

Literature Survey

In this chapter we explore previous work related to this thesis that has both inspired and informed our research. This places the research presented in this thesis within the broader context of agents, games, adaptation, and personality theories. We begin with an overview of some of the concepts used in our model that are explained in greater detail in Chapter 3. This overview of our model will motivate why we present only certain theories in this chapter. After this overview, we introduce the theories pertinent to our research, followed by applications of these theories.

Our model of personality draws from cognitive-social theories that suggest one way we learn is via self-reinforcement based on past experience. Hence, past experience forms a core part of personality in our model. This past experience influences decision-making, so that the choice of what to do depends on the success of what was done last time this context was perceived. Damasio's somatic marker hypothesis (Damasio, 1994) offers an explanation of how past experience influences decision-making. Somatic markers act as context-dependent preferences that guide decisions towards or away from specific actions or choices.

Characters use the Beliefs, Desires, Intentions (BDI) agent paradigm to represent how they reason. We use the BDI paradigm because it relates explicitly to goals, success and reasoning, and uses terminology that allows behaviour to be explained easily. An adaptation loop or learning loop is integrated into the standard BDI execution loop so that characters can develop their own personalities. To generate and update somatic markers, characters use a learning loop and calculate a self-reinforcement value or personal reward and use a simple reinforcement learning technique from Sutton &

Barto (1998), called the reinforcement comparison technique (see Section 2.1.4.2, page 40). An *agent* (the reasoning part of a character) uses learning to determine which action is the most appropriate choice, given its current personal context. Based on the character's experience, it builds up a database of which actions it prefers above others for a given context, i.e. it builds up its somatic markers. Character personality is visible in the choices the character makes between activities and the sub-plans it chooses to execute its chosen action.

After a character has completed an activity, it then evaluates the success or failure of this activity at helping it achieve its own overall goals. This self-evaluation allows the character to update its preferences, i.e. the character uses self-reinforcement. In addition to influencing choices made, personality also influences how to evaluate the success or failure of an activity. For example, one character may consider having a lot of money as success whereas another may want to have a lot of friends. To model this other aspect of personality in our model, we use *soft goals* to evaluate success or failure of completed activities. Soft goals are goals that should be achieved, but the character initially has no explicit knowledge of how to achieve these goals.

Based on the introductory chapter and this brief overview, the key theories that our model draws from are: agents, personality theories, somatic marker hypothesis, and adaptation or learning techniques. We begin the literature survey by introducing these theories. This is then followed by examining applications from games and intelligent virtual agents that use these theories and inform our model.

2.1 Theories

In this section of the literature survey we discuss theories and techniques relevant to this thesis, as well as theories used by other applications in the field of intelligent virtual agents and computer games. We begin by explaining theories and methods relating to *agents*. Then we discuss psychological and cognitive science theories of personality. The background to Damasio's somatic marker hypothesis is explained, followed by a discussion of techniques used for adaptation and learning. In particular, reinforcement learning is discussed as this is relevant to enable characters to adapt their behaviour preferences.

2.1.1 Agent Theories

In this section we describe theories, techniques and definitions that are used for "agents" and intelligent virtual agents (IVAs). There are a variety of definitions of an agent in the literature. We use the term "agent" to refer to the reasoning part of a character, rather than the visual appearance. Agents are rational and model human behaviour. A common position, that we adopt, is that agents have the following properties (from (Padgham & Winikoff, 2004)):

- situated: exist in an environment;
- autonomous: behave independently and not controlled externally, i.e. they make their own decision on which actions to implement;
- reactive: respond in a timely manner to changes in the environment;
- proactive: persistently pursue goals;
- flexible: have multiple ways to achieve goals;
- robust: recover from failure;
- social: interact with other agents.

We begin this section by explaining the beliefs, desires, intentions (BDI) agent paradigm that models how agents reason about the world. This is followed by an exploration of the different goal types that are used in the literature and within this thesis. We then explain the core aspects of the cognitive appraisal model of emotions that is used by many intelligent virtual agents applications. We finish by describing methods to measure believability of characters in games and virtual worlds.

2.1.1.1 Beliefs, Desires, Intentions (BDI) Agents

This thesis uses the Beliefs, Desires, Intentions (BDI) paradigm of agent programming. In this paradigm, based on work by Rao & Georgeff (e.g. Rao & Georgeff, 1995), agents function in a manner similar to the way people normally reason about themselves. This makes it easier for designers to understand and therefore debug characters, as well as making it easier for players to understand why a character behaves the way it does. BDI techniques map well to problems where there is no clear solution (Norling, 2004), such as games where there are multiple ways to achieve the same goal. In the BDI paradigm, an agent stores beliefs or knowledge about themselves and their environment. The agent also has a number of desires that represent states it is trying to achieve. Desires can

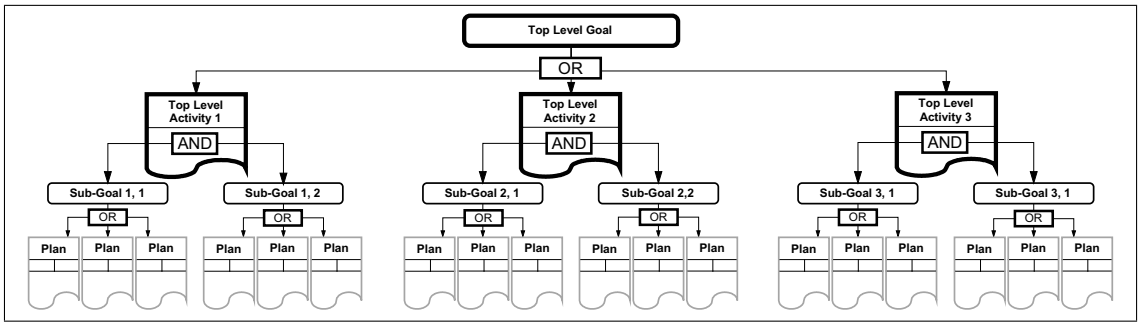

Figure 2.1: Generic goal/plan hierarchy: each goal can be handled by one of three plans. Each plan is implemented by achieving two sub-goals.

also have payoffs associated with them (Rao & Georgeff, 1995), e.g. some desires are considered more important than others. The set of desires that the agent is currently committed to achieve are termed goals. Goals are a subset of desires and must be able to be achieved simultaneously. Whereas the desires of an agent may contain conflicting goals (Thangarajah *et al.*, 2002). In Krümpelmann *et al.* (2008), a motivation factor induces a pre-ordering of desires, so that an agent is able to choose a single goal to pursue at a time.

Once the agent has chosen a goal to attempt to achieve initially, it forms an intention, or plan, to achieve it. Plans represent ways that an agent can achieve their goals (and consequently desires). Plans have an invocation condition to specify the triggering event (relevant goal) that the plan handles. Plans also can have a precondition that specifies the situation that must hold for the plan to be executable (Rao & Georgeff, 1995).

For example, an agent, called Gina, believes she is not talking to anyone currently and is not reading. She desires to talk to someone and also to read a book by herself. Based on her reasoning model, she chooses the goal to talk. She cannot simultaneously choose to talk *and* to read a book, since she is not able to do both at the same time. Since she has chosen to achieve the goal "talk", she must now choose how she will achieve this goal. So, based on her reasoning model, she chooses a plan to achieve the goal of talking to someone, such as the plan "have a conversation".

Agents designed using the BDI paradigm have a number of goals that they can achieve, as well as a number of different plans that can achieve these goals. The designer explicitly links plans to the goal they handle, and specifies whether plans

require further sub-goals to be achieved. These links between goals and plans are usually represented in a goal/plan hierarchy. Figure 2.1 shows a generic version of a goal/plan hierarchy according to BDI methods. In the hierarchy shown, the agent has a top-level goal it wants to achieve. It can do this by implementing any of the three available plans or activities. Each plan has two sub-goals that must both be achieved for the plan to succeed. In turn, each sub-goal can be achieved by choosing one of three plans. For example, if the top-level goal is to have a conversation, the agent can do this by choosing from three plans: talking to a friend, or an enemy, or someone they have not met before. Once they have chosen to talk to a friend, they would need to achieve the goals of choosing what to say and ending the conversation. Note that in real-world examples the goal/plan hierarchy developed is not usually as symmetric as our example.

By structuring goals and plans into this hierarchy, the designer is able to provide the agent with a large number of ways to achieve its top-level goal. If the goal/plan hierarchy has a depth of D (based on number of goal levels), always has C plans applicable for each goal, and S sub-goals for each plan, then the number of ways in which a goal at the top of a goal/plan hierarchy can be achieved is (Padgham & Winikoff, 2004):

$$C^{(\frac{S^D-1}{S-1})} \tag{2.1}$$

In the generic goal/plan hierarchy in Figure 2.1, $C = 3$; $S = 2$ and $D = 2$, so the number of ways that the top-level goal can be achieved is $3^{(\frac{2^2-1}{2-1})} = 27$. This enables greater variety in behaviour without requiring these paths to be coded explicitly.

In the BDI paradigm, each agent uses a standard execution loop (d'Inverno *et al.*, 2004; Rao & Georgeff, 1995) to act within the world, see Figure 2.2. Goals are usually represented as events in many BDI implementations. An event is a goal that is sent by a plan or an agent and once handled by an applicable plan the event is removed, i.e. events are usually not persistent. The loop begins with the agent observing the world and its own internal state to determine whether there are any new, incoming, events. The event queue is updated with this information. The next event is taken from the event queue and the agent chooses a plan to execute using its beliefs and goal/plan hierarchy. The set of available plans is constructed based on whether the plan is applicable, i.e. will handle the event being considered and is valid in this world state. For example, Gina cannot choose the plan "talk to an enemy" if she does not

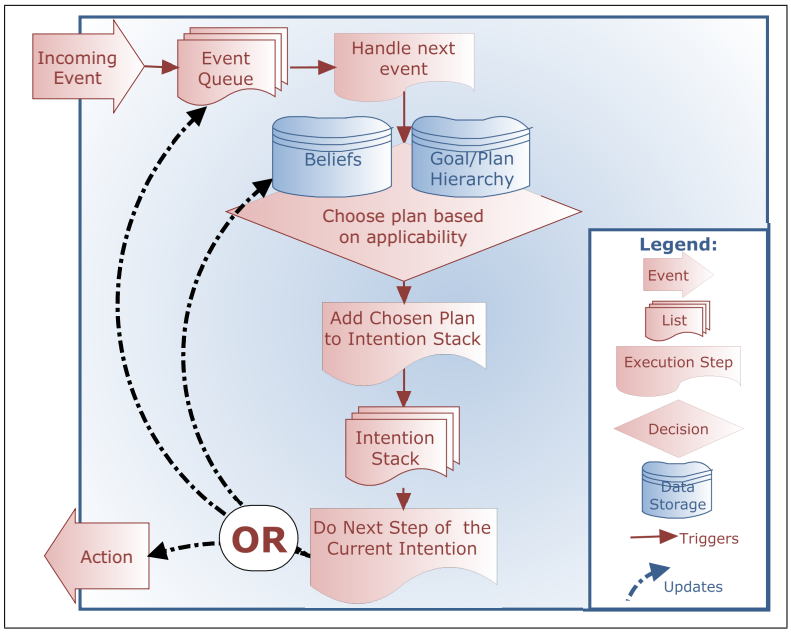

Figure 2.2: Standard BDI execution cycle.

currently have any enemies. The chosen plan is pushed onto the intention stack. The next step of the plan that is at the top of the intention stack will be executed. This step may involve changing the agent's own beliefs, generating a new event (for itself of sending an event request to another agent), or acting in the environment itself. After this, the loop begins again and continues while the simulation is running.

Modelling agents to have beliefs, desires and intentions using the BDI paradigm is a way of representing and generating agent behaviour that is easy for people to understand, since it is often how we explain our own behaviour. Existing programming languages, such as the JACK programming language, have automatic support for the BDI paradigm (Howden *et al.*, 2001).

2.1.1.2 Goal Types and Motivations

In the BDI paradigm there are a number of different goal types (Huber, 1999; van Riemsdijk *et al.*, 2008). Declarative goals are goals "to be", where the agent wants to reach a certain state of affairs, e.g. "have ten baked bread rolls". Procedural goals are goals "to do", when the agent wants to execute actions (van Riemsdijk *et al.*, 2008), e.g. "bake bread". Within these main types there are number of sub-types of goals,

including achieve, maintain and perform. Achievement goals are where the agent wants to be in a specific state, e.g. "be at the bakery", whereas maintenance goals are a state that the agent wants to maintain over a period of time, e.g. "do not be hungry". Perform goals are a set of actions that the agent would like to do, irrespective of their potential outcome, e.g. "go for a stroll". Goals are usually dropped once they are performed, achieved or maintained for the required duration.

Given the less precise nature of personality and to reduce the designer's burden, we should consider goals that may not have an explicit plan that achieves them. The model of emotions by Ortony, Clore and Collins (OCC) (see Section 2.1.1.3, page 27) uses emotional goals that link well to personality models. According to the OCC model of emotions, there are three types of goals people have in the real world: Active-pursuit goals, Interest goals, and Replenishment goals (Ortony *et al.*, 1988). Active-pursuit goals (A-goals) are goals that a person tries to obtain, such as become a baker, or engage in a conversation. They also represent things one wants to get done, like bake bread. Interest goals (I-goals) are goals that are usually not pursued actively, because one has little control over their realisation, such as preserving one's health or that one's friends should prosper (Ortony *et al.*, 1988). I-goals are situations one wants to see happen. Replenishment goals (R-goals) are goals that wax and wane, such as hunger and getting petrol for one's car. R-goals are somewhat similar to maintenance goals that are sometimes actively pursued, and other times simply monitored for failure.

In many systems, the types of supported goals are very functional such as: "bake bread", "engage in conversation". These goals are easily achievable by implementation of plans such as: " bake sourdough bread", "bake white bread". Higher level A-goals are less clear cut. For example, "have friends" could be achieved or partially achieved by talking to someone or giving away food or other choices depending on the domain. In some instances this may be explicitly coded by the designer, but to reduce designer workload it makes sense to make a distinction between the low-level functional goals and higher level goals that cannot be directly achieved in the goal/plan hierarchy. The term "soft goals" is used to refer to these non-functional goals that do not have explicit plans to achieve them (Braubach *et al.*, 2004).

Soft goals are distinct from, but related to, motivations which are also used in agent and planning research (Coddington & Luck, 2004; Norman, 1994, 1997). Work by Coddington & Luck is applied to a planning domain, but uses similar terminology to

Norman's BDI-based agent research to improve goal management. Motivations allow a planner or an agent to consider time and resources in addition to the traditional planning analysis of number of actions and outstanding goals (Coddington & Luck, 2004). Motivations reflect the drive of an agent and are used to directly generate goals and affect plan evaluation. The current values and importance of each motivation are linked and change in relation to physical environmental changes (Coddington & Luck, 2004). Unlike soft goals, which have both a current value and a separate importance value, if all motivations have the same value, then they are considered equally important to achieve. When one particular drive is not being achieved, the agent will generate goals that will actively improve that particular motivation value. When a goal is generated it is given a priority based on time-related deadlines as well as the current strength of the motivation that generated it. For example, if the agent is *very* hungry now, a goal to find food will be given higher priority, compared to if the agent is just mildly hungry. These priorities values are used to determine which goal to trigger and pursue next (Coddington & Luck, 2004; Norman, 1997).

Soft goals are high level goals that are more general than hard (standard) BDI goals or motivations that have a clear way to achieve them. Initially, the agents have no knowledge of how to achieve these soft goals and so they must learn, via trial and error, which plans allow them to achieve or progress towards achieving their soft goals. That is, the main way that characters will adapt is to learn how to achieve their soft goals simultaneously. The soft goals that an individual agent is trying to achieve depend upon its personality. Soft goals act somewhat like maintenance goals; although an individual soft goal may be achieved, the agent will not drop the goal. It will continue to ensure that its actions do not cause the goal to fail in the future. That is, we assume that once the agent is rich, it wants to stay rich. An agent does not seek to achieve its soft goals separately, rather they are trying to achieve all of them simultaneously. Some soft goals may be more, or less, important than others and therefore their perceived proximity to achieving all goals will be higher when all the important goals are achieved, compared to when the less important goals are achieved.

Comparing soft goals and motivations we note that, although both represent high-level goals the agent wishes to achieve, motivations are quite functional and usually relate to essentials that the agent *must* achieve or satisfy Coddington & Luck (2004); Norman (1997), such as health or resources. Whereas soft goals relate to states that we

would like to be satisfied (some more than others), but can physically live without, e.g. having friends. Unlike soft goals, motivations have explicitly linked goals that they can generate when they are not achieving a particular motivation. Further, motivations use their current value to directly give an instantiated priority to specific generated goals and actions. Although a prediction of the improvement to soft goal achievement values is used to determine preference (or priority) of actions, these preferences are not explicitly programmed, but are learnt by the agent by trial and error.

2.1.1.3 Cognitive Appraisal Model

The cognitive appraisal model is used in many intelligent virtual agent applications (e.g. André *et al.*, 1999; Dias *et al.*, 2005; Egges *et al.*, 2004; Gratch & Marsella, 2004). There are many variations on cognitive appraisal, but the main premise is that emotions can only be updated or triggered after an appraisal of the world and events. In other words, before an emotion is felt, a cognitive process is necessary so that incoming events can be interpreted and meaning attached to them. For example, a dark alley can cause fear if one remembers, perhaps subconsciously, a reason to be afraid (such as watching a scary movie recently). Two of the most influential works are the models by Ortony, Clore & Collins (1988) (OCC) and Lazarus (1991).

In Lazarus's model (1991) an incoming event triggers an *appraisal* that then leads to the person implementing a *coping* strategy to deal with the event. Coping relates to how to think and deal with emotional encounters and appraisal relates to how to interpret events and what strategy to use to cope. There are three types of appraisal according to Lazarus (1991):

1. Primary appraisal: occurs when an incoming event is received. This process analyses the event to determine the relevance to the person's well-being.
2. Secondary appraisal: chooses between coping choices in order to determine how to deal with emotional encounters.
3. Re-appraisal: an evaluation of feedback from the environment based on one's own actions and reactions.

Primary appraisal is the key to how emotional responses differ or are the same. If two individuals appraise different situations in the same way, their emotional response will

be the same, but "if two individuals appraise the same situation differently, their emotional response will differ" (Lazarus, 1994, p.336). After an appraisal, the person deals with the result of the appraisal via coping using: *problem-focused coping* or *emotion-focused coping* (Lazarus, 1991). Problem-focused coping processes are generally any form of behaviour that the agent is able to exhibit in the virtual environment, such as gestures and actions. For example, a person who is unhappy because they do not have a car, can work to be able to buy a car. Alternatively, the person may modify their values so that not having a car is something to be proud of. The emotion-focused coping process captures this second kind of mechanism and can change beliefs, desires and intentions.

The "OCC" model was first proposed in Ortony, Clore & Collins (1988). Its main focus is its investigation of how to break down the primary-appraisal process into parts to describe how different emotions are generated and which variables influence the appraisal process. Appraisal depends on goals, standards and attitudes (Ortony *et al.*, 1988). Variables that influence which emotion is triggered include desirability, praiseworthiness and appealingness (Ortony *et al.*, 1988). The intensity of the emotion generated depends on both local and global variables, such as reality, proximity, unexpectedness, arousal, likelihood, and deservingness (Ortony *et al.*, 1988).

Cognitive appraisal models require substantial world and individual models to be developed so that incoming events can be appraised appropriately to generate emotions for each person and for every possible situation. Otherwise a method to generalise events would be needed. As a minimum, to implement primary appraisal in the OCC model, the designer needs a model of expectations (Seif El-Nasr *et al.*, 1998), a method to determine what the event means to the character and a goal hierarchy to calculate desirability (Bartneck, 2002). Ortony himself later described the OCC model as "the rather cumbersome (and to some degree arbitrary) analysis" (Ortony, 2002, p.193).

2.1.1.4 Measurement Techniques for Believability

One common approach to determining "success" of virtual characters and their model is to rate how believable the characters are. Although there is discussion about the need for measures other than believability (Gratch & Marsella, 2005), many applications would still like to achieve a high level of believability of their characters and in some cases realism. In order to measure or evaluate the subjective quantity of believability,

an audience is needed (Mateas, 1997). People can find believability and personality hard to judge and this is commonly due to lack of expressiveness of agents (Jan & Traum, 2005) or other visual problems.

The dream list of what an intelligent virtual agent (IVA) should have to be believable or project the "illusion of life" is commonly thought to include personality, emotion, relationships, making its own decisions, have roles, follow social conventions, respond with empathy, be self-motivated, change (grow and change with time, in a manner consistent with their personality), and an illusion of life that includes pursuing multiple, simultaneous goals and actions (Hayes-Roth & Doyle, 1998; Mateas, 1997), self-perception and self-esteem (Seif El-Nasr *et al.*, 1999), reactive, situated and embodied behaviour (Mateas, 1997), realistic (for real-world simulations) (Johns & Silverman, 2001), and not be entirely predictable (Henninger *et al.*, 2003).

Ruttkay, Dormann & Noot (2002) proposed a framework to compare embodied conversational agents (ECAs) to each other and to traditional input methods. ECAs are usually a "talking head" on a screen that interacts with a user, often within a functional application such as providing tourist information. The framework of Ruttkay *et al.* (2002) is a series of mostly subjective questions relating to the design of the character as well as how to evaluate the character itself. The possible methods of collecting empirical data are observation of users, experiment (where users are involved as subjects in a controlled way), criteria and comparative tests, survey and online survey, questionnaire, interview, focus group, and usage data (Ruttkay *et al.*, 2002). Questions cover aspects of the character including actual embodiment, representation of the mind, how users control or interact with the character, ease of use, user satisfaction, trust, and engagement (Ruttkay *et al.*, 2002). Despite the breadth of this framework, it mostly relied on non-quantifiable or subjective questions such as "In what way does the model of the user influence the communication of the ECA?" (Ruttkay *et al.*, 2002, p.3) or "Is the user pleased with using the ECA?" (Ruttkay *et al.*, 2002, p.6). Another framework to compare characters in virtual environments (particularly military simulations) can be found in Sandercock *et al.* (2004). In both frameworks the subjective nature of the questions makes it difficult to compare applications or eliminate participant biases. Further, these frameworks do not address questions relating to the choice and number of subjects. Some studies have used less than ten questionnaire participants (e.g. Jan & Traum, 2005; Rousseau & Hayes-Roth, 1997) and this seems unlikely to be able to

establish statistical significance, particularly in light of the large number of questions asked of the participants.

Turing Test A classic measure of artificial intelligence (AI) is the Turing Test (Turing, 1950). The original Turing Test proposed by Alan Turing related to distinguishing a woman from a man and then whether a machine could be distinguished from a woman (Rousseau & Hayes-Roth, 1997). Although this test is used less frequently in recent times, game AI researchers have advocated its use, particularly for computer-controlled opponents in first person shooters (FPS), called Bots (Glende, 2004; Laird & Duchi, 2000; Livingstone, 2006; MacInnes, 2004; Sandercock, 2004). The Turing Test places an emphasis on the appearance of intelligence and does not constitute proof that the computer character actually *is* intelligent (Livingstone, 2006). This appearance of intelligence is similar to the aim of believability of characters.

MacInnes (2004) used the Turing Test in a custom-built FPS game where opponents (Bots) were created using different AI techniques; finite state machines (FSM), neural network and "Mixture of Experts". Laird & Duchi (2000) used a Turing Test to assess custom Bots in *Quake* (by id Software) to determine "humanness" and which parameters affected perception of this.

Both Sandercock (2004) and Livingstone (2006) used the Turing Test to look for weaknesses in Bot believability, in order to determine ways Bots can be improved. Livingstone (2006) believed that a questionnaire is more effective when it presents participants with two versions of a character and asks which is more believable, because this is likely to decrease problems with some participants who always say "no" or "yes". Livingstone (2006) and Sandercock (2004) used extensive surveys to determine how people made their decisions on whether the opponent was human or artificial.

When people know they are being asked to look for a Bot or a specific number of them, their responses may be biased (Sandercock, 2004). If the participant is unaware that a character may not be human, then they may not notice that it is a Bot (Livingstone, 2006; Sandercock, 2004). In Sandercock's study, to eliminate this biasing effect, participants played a number of different games where the number of human-controlled opponents versus computer-controlled Bots was varied without the participant's knowledge (Sandercock, 2004).

2.1.2 Personality Theories

According to Ortony (2002), personality should be viewed as a driver of behaviour. A key component of development, both emotional and otherwise, is an individual's acquisition of the personality characteristics that influence all types of appraisal and coping (Lazarus, 1991). This acquisition process can be viewed from both the perspective of innate tendencies (nature) and variable experience (nurture) (Lazarus, 1991). In this section, we describe common theories of personality and discuss their varying approaches.

Personality theories attempt to understand and describe why each person is, in certain respects, like all other people, like some other people and yet like no other person (Kluckhohn & Murray, 1953). That is, all people are born and are part of the world as are all other people; but there are common traits or similarities that can be noticed amongst specific individuals or groups (Kluckhohn & Murray, 1953). However:

> "The ultimate uniqueness of each personality is the product of countless and successive interactions between the maturing constitution and different environing situations from birth onward. An identical sequence of such determining influences is never reproduced" (Kluckhohn & Murray, 1953, p.55).

In this section, we discuss two of the many existing personality theories: trait-based and cognitive-social. Although other approaches may be equally valid, trait-based theories are used frequently in virtual agent applications and games, and cognitive-social theories offer an explanation of how personality is developed in a way that could be implemented in a virtual agent. According to Ortony *et al.* (2005), there are two main methodologies to analysing personality and individual differences; the first seeks to identify the dimensions by which we differ from each other, the second questions how personality affects deeper functioning and how it is developed . The first methodology can lead to trait-based approaches, the second methodology can lead to cognitive-social based approaches, which offer an explanation of how personality develops.

We begin by considering trait-based theories and identify their deficiencies. Then we introduce cognitive-social theories. Finally, we present a section on individual differences: reasons for behavioural and personality differences between and within individuals.

2.1.2.1 Trait-based Personality Theories

Trait-based theorists assume that people display "broad predispositions to behave in particular ways" (Cervone & Pervin, 2008, p.236). These theories describe or label personality types based on what we can observe in others (Ortony *et al.*, 2005). For example, describing a person as extroverted, shy, or aggressive. Personality traits are identified as "consistent patterns in the way individuals behave, feel, and think" (Cervone & Pervin, 2008, p.238). Trait-based theories assume that an individual's tendencies are more important than the situation they are in (Pervin *et al.*, 2005), that average levels of behaviour are more important than patterns of variability in action (Cervone & Pervin, 2008). In reality, "learning can occur throughout life" (Cloninger, 2008, p.343), behaviour *can* change to meet needs and goals and personality itself can change over extended periods of time (Pervin *et al.*, 2005). Trait-based theories do not provide an explanation to address these issues of personality development (Pervin *et al.*, 2005).

Trait-based theories are frequently used when constructing intelligent virtual agents (IVAs) and characters in games (see application sections: for games see Section 2.2.1.1, page 42; for IVAs see Section 2.2.2.1, page 51). Trait-based theories generally require construction of a schema of key personality dimensions and these schema can be classified according to the number of dimensions chosen. Many rely on three key dimensions, but there are several popular versions using more, for example, the Myers-Briggs type indicator (four dimensions) (Myers & McCaulley, 1985) and the five-factor model (McCrae & John, 1992). When implementing a trait-based theory in a virtual world, the designer must consider in detail how each dimension or trait affects behaviour, reasoning and appearance, then set up individual characters based on some combination of values for each dimension. The character cannot change its traits over the course of the game, even when there are on substantial changes to the environment. This deficiency is addressed in cognitive-social theories.

2.1.2.2 Cognitive-Social Theories

Cognitive-social theorists believe that personality is acquired based on experiences with the environment; and behaviour is due to the effect of environment on the person (Pervin *et al.*, 2005). Adult personality may generally considered to be static and

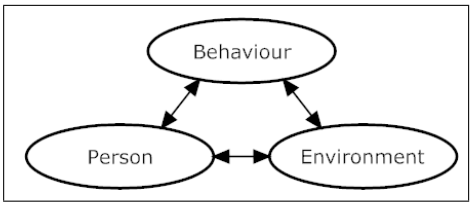

Figure 2.3: Reciprocal determinism in cognitive-social theories: how behaviour, the person and the environment influence each other (adapted from Bandura (1977)).

does not adapt or change over time; therefore appearing to be suitable to trait-based approaches to personality. Within virtual domains, this assumption may be appropriate if the personality model is complex enough to begin with, but this requires handcrafting each character, or setting up several personality types. However, adopting a cognitive-social view of personality allows the characters to develop by themselves and generate more complex personalities. That is, the characters can simulate in some respects the way personality develops in childhood and adolescence.

People are key to cognitive-social theories (Cervone & Pervin, 2008). People can reason about the world, the past and the future as well as reflect about themselves (Cervone & Pervin, 2008). They are in control of their own actions and can motivate and direct their own actions (Cervone & Pervin, 2008). Behaviour results from the complex interaction of persons and the environment, rather than from any single factor alone, see Figure 2.3 (Bandura, 1977). That is, people are neither driven by inner forces nor buffeted by environmental stimulus (Bandura, 1977). The traditional view of behaviour interaction is that a person's behaviour is a function of the person and the environment. However, people's actions and behaviour contribute to the overall environment (Bandura, 1977). The overall environment will affect the experiences a person has and what they become, and also their subsequent behaviour (Bandura, 1977). This mutual influence of the person, the environment, and behaviour is called reciprocal determinism. According to cognitive-social theorists, the behaviour and cognitive processes of individuals are different due to their learning process (Cloninger, 2008). Further, situations can be linked to different sets of cognitions and effects, and behaviour chosen based on different situations (Cervone & Pervin, 2008).

In cognitive-social theories, "cognitions about what the world actually is like (beliefs), about one's aims for the future (goals), and about how things normatively should

be (standards) play distinct roles in personality functioning" (Cervone & Pervin, 2008, p.469). This is similar to the BDI paradigm (see Section 2.1.1.1, page 21). The key concepts in cognitive-social theories are listed below.

- Competencies and skills: people can do different actions differently and context is important (Cervone & Pervin, 2008).
- Beliefs and expectancies: what the world is like and what it probably will be like in the future (Cervone & Pervin, 2008).
- Behavioural standards: people acquire different criteria for evaluating events, called self-evaluative reactions (Cervone & Pervin, 2008). These evaluations influence our future actions and emotions by making us "respond in an emotionally satisfied or dissatisfied way toward ourselves" (Cervone & Pervin, 2008, p.467).
- Personal goals: people can envision the future, therefore they can make specific goals for actions and can "motivate and direct their own behaviour" (Cervone & Pervin, 2008, p.464).

Learning Types in Cognitive-Social Theories Traditional learning occurs by taking action and experiencing the effects. According to Bandura (1977), a large part of learning occurs from observing other people's behaviour and the consequences for them, rather than for the person who is learning. He describes three main types of learning.

1. ***Learning by Response Consequences***

   Informative and Reinforcing Function: Observe the outcomes of your own actions and use this as a guide for future actions. This can only reinforce behaviour if the reward/punishment is linked to that behaviour. If the individual does not know what is being punished, then behaviour cannot change (Bandura, 1977).

   Motivational Function: Past experience allows the individual to create expectations that certain actions lead to benefits, have no appreciable effect or maybe will avert trouble. These foreseeable outcomes can become motivators of behaviour (Bandura, 1977).

2. ***Learning through Modeling***

   Observe others and from this form an idea of how new behaviour is formed and subsequently use this later to guide action (Bandura, 1977).

3. ***Self-reinforcement***

   This type of learning relates to how behaviour is regulated by the interplay between self-generated and external sources of influence. Performance improves mainly via the motivational function linked to the self-regulated reinforcement (Bandura, 1977). The self-regulation process is a self-observation, then a judgmental process followed by the self-response (Cloninger, 2008). The reinforcement value is based on how much the individual (not an external trainer) prefers one outcome over another (Cloninger, 2008; Phares & Chaplin, 1997). Behaviour is evaluated partly based on how others react to that behaviour (Bandura, 1977).

2.1.2.3 Individual Differences

Personality can be described as "a generative engine that contributes to coherence, consistency, and predictability in emotional reactions and responses" (Ortony, 2002). Our unique personalities can cause each of us to react differently, even when responding to the same provoking situation. Also, the same individual can react differently depending on the situation. For example, someone in an aggressive environment is likely to be more aggressive, but this same person may be very calm in another environment (Ortony, 2002). Both Ortony (2002) and Lazarus (1994) have addressed the possible reasons why these individual differences occur.

According to Ortony (2002), individual differences are due to:

1. differences in evaluation and construal of the world (e.g. whether you are winning a football match or not depends on which team you are on; and importance placed on winning affects evaluation);
2. differences in the way that emotions affect us, called *emotionality* (e.g some people are more volatile than others); and
3. current state of the individual and their view of the environment.

On the other hand, according to Lazarus (1994, p. 334), different reactions to the same provoking situation are due to "variable individual goal hierarchies, generalised beliefs about self and world, and situational beliefs", as well as environmental differences. How one deals with events or how one acts to change their beliefs or actions also generates individual differences (Lazarus, 1994).

According to Caspi & Roberts (1999), there are number of ways that differences in personality can be measured.

1. Differential Continuity: change in an individual's placement relative to the group.
2. Absolute Continuity: change in the quantity or amount of an attribute over time.
3. Structural Continuity: persistence of correlation patterns among a set of variables across time.
4. Ipsative or Person-centred Continuity: change at the individual level, or the configuration of variables within an individual across time.
5. Coherence: refers to conceptual rather than literal continuity among behaviours. An example of this type of coherence is relating behaviour and attributes as a child (aggression, social nature, physical adventurousness and nonconformity) to adult sexual behaviour (Caspi & Roberts, 1999).

2.1.3 Somatic Marker Hypothesis

In the somatic marker hypothesis proposed by Damasio (1994), he rejects the belief (held by Descartes, amongst others) that the mind and the body are separate entities. Damasio believes that when making decisions our feelings or bodies assist us in an indispensable way. When faced with a decision with many choices, the individual may experience an unpleasant physical reaction, or gut instinct, in relation to one or more of the choices available (Damasio, 1994). This will cause the individual to immediately view those choices as negative and encourage them to choose from the other alternatives. This type of physical reaction (feelings) are called a "somatic markers" because it is a bodily feeling ('soma' means body) and 'marks' an image or choice (Damasio, 1994).

When choosing between courses of action, choices can be bucketed using somatic markers to establish preferences (Damasio, 1994). The internal preference system is inherently biased towards avoiding pain and seeking potential pleasure (Damasio, 1994). Somatic markers represent, at any given time, the cumulative preferences a person has received and acquired. Somatic markers act as biasing devices: a negative somatic

marker is like an alarm bell; a positive one is like a beacon of incentive (Damasio, 1994). Somatic markers do not deliberate for us, they highlight choices for the deliberation process. That is, they drastically reduce the number of choices that need to be examined if further cost/benefit analysis is required. According to Damasio, the accuracy and efficiency of the decision process is increased with somatic markers (Damasio, 1994). In some cases, such as intuition, somatic markers are formed and used unconsciously without recognising their existence.

Somatic markers come from our experiences and socialisation (rather than our genetics), and are largely acquired during childhood and adolescence. However, the acquisition of somatic markers continues throughout our entire lives (Damasio, 1994). The person must connect entities or events with the enactment of a body state, pleasant or unpleasant. Somatic markers are acquired by experience, under the control of an internal preference system and under the influence of social conventions, ethical rules and the other entities with which a person must interact (Damasio, 1994).

By their very nature, somatic markers are dependent on the context in which the action possibilities are being considered. The hypothesis is a useful way of representing how agents can make decisions without domain-dependent deliberation. It provides a simple structure to allow preferences, personality and intuition to influence current decisions.

2.1.4 Adaptation Theories

As humans, people are continually adapting to the environment, mostly because the environment is in a continual state of change. People acquire new goals and beliefs as they age. However, it is generally assumed that the most important and stable goal hierarchies and beliefs are established during our formative years before adulthood (Lazarus, 1991). Further, learning can give the appearance of personality (Sanchez *et al.*, 2004).

Our focus is on virtual agents, not the ways people in the real world actually learn. So, in this literature survey we examine simple machine learning techniques that can be used to allow personality to adapt and develop via personal experience. We look first at the aspects that can be learned, then we outline the main concepts of reinforcement learning and finally concentrate on the particular learning technique that will be used in this thesis, reinforcement comparison.

2.1.4.1 Aspects that can be Learned

Agents in virtual worlds can acquire knowledge about a multitude of aspects of their environment, including the other inhabitants and themselves. Aspects open to learning can be categorised as follows:

- Concept learning about objects and other characters: (Seif El-Nasr *et al.*, 1998; Yoon *et al.*, 2000) which objects and characters help achieve goals? Which objects are associated with certain motivational states or emotions (Seif El-Nasr *et al.*, 1998).
- Social learning: what other characters are like (in terms of their behaviour and likes and dislikes), when to collaborate, when to compete.
- Organisational learning: includes updating the relative importance (or weighting) of the connections between entities as well as changes in the structure of an organisational network (Yoon *et al.*, 2000).
- Preferences for actions or strategy: which action or strategy is "good" (i.e. preferable) in a particular situation. This can be related to forming somatic markers based on Damasio's somatic marker hypothesis (Yoon *et al.*, 2000).
- Learning about events: likelihood of events to occur at any given situation, which events are "good" (i.e. which states should be achieved, compared to learning which actions are good), event sequences, and potential consequences and rewards (Seif El-Nasr *et al.*, 1998).
- Learning about the human user: (Seif El-Nasr *et al.*, 1998) what does the user like (actions, objects, etc...)? What is the user's current emotion, mood or personality?

Reinforcement learning is a relatively simple method of machine learning and has been used commonly to learn the above aspects.

2.1.4.2 Reinforcement Learning

Reinforcement learning (RL) is derived from animal training techniques where the animal is given a reward based on its good or bad behaviour. The goal of reinforcement

learning is to maximise reward by mapping situations to actions, i.e. what to do in a given situation. Usually the reward is externally determined by a training agent that is separate from the agent that is learning. Through trial and error interaction with the training agent, the learning agent is able to acquire knowledge about what are "good" and "bad" states (according to the external trainer) and which actions or behaviour lead to "good" states and therefore rewards.

There are four main elements in a RL system (Sutton & Barto, 1998): selection policy, reward function, value function and the model of the environment. The selection policy is the function that maps perceived states of the environment to action. The selection policy needs a mechanism to handle the trade-off between exploration of all state-action pairs and exploitation of known successful state-action pairs (Sutton & Barto, 1998). Some policies only exploit successful state-action pairs without exploring further. These policies are called greedy policies. ϵ-greedy policies exploit the successful state-action pairs only some of the time, based on the parameter ϵ. The reward function should be unalterable by the agent and clearly related to the pre-acquired goal of the agent. The reward function is required to map the state of the environment to a single number - the reward. The value function defines what is "good" in the long run for the agent, e.g. getting high rewards is good. The model of the environment mimics the behaviour of the environment. If a particular action is taken when in a specific state, the model of the environment can predict the next state and the next reward. To simplify this process, the actual virtual environment can be used instead of a model of the environment could do. Techniques that use this method include both simple techniques, such as reinforcement comparison, and more complex techniques, such as Q-learning.

Properties of Virtual Environments According to Russell & Norvig (2003), virtual environments can be categorised based on four properties: observability, determinism, dynamics, and the number of agents. The properties of most game environments are likely to be:

- partially observable: the agent cannot determine the state of the environment fully;
- non-deterministic: the next state of the environment is not completely determined by the current state and the agent's actions;

- dynamic: the environment can change while the agent is deciding what to do; and
- multi-agent: other agents can affect the state of the environment.

Many RL techniques assume the environment is deterministic, thus making them difficult to implement in games. However, one simple technique that does not require the agent to already have a model of how the environment behaves, is the reinforcement comparison technique.

Reinforcement Comparison Technique The reinforcement comparison technique provides a mechanism to update the selection policy based on the reward, without requiring a complex model of the environment (Sutton & Barto, 1998). It determines whether a reward is "large" or "small" based on previous rewards received. In this process, a reference reward, $\bar{r}_t$, (usually an average of previous rewards) is stored to provide a comparator for future rewards. The updated preference, $p_{t+1}(a_t)$, for an action, a_t, selected on the last play is (Sutton & Barto, 1998):

$$p_{t+1}(a_t) = p_t(a_t) + \beta(r_t - \bar{r}_t) \tag{2.2}$$

where r_t is the reward received on the last play and β is a positive step-wise parameter. To update the reference reward, the following equation is used (Sutton & Barto, 1998):

$$\bar{r}_{t+1} = \bar{r}_t + \alpha[r_t - \bar{r}_t] \tag{2.3}$$

where $0 < \alpha \leq 1$. If the initial reference reward, $\bar{r}_0$, is set at a high level, then this equation encourages exploration.

The reinforcement comparison technique is expected to work well to match with a personality theory, since according to Moffat (1997), personality theories require that the reinforcement value should reflect the expectancy value. That is, determining whether the result is good or bad depends on whether the agent was expecting a good or bad result to begin with (Moffat, 1997).

2.2 Applications

Having surveyed the theories and methods most relevant to our topic, we now investigate applications that others have implemented. These applications are separated into

those intended for use in computer games and those in the broad field of intelligent virtual agents. Some applications incorporate theories of personality, adaptation and somatic markers. However, to facilitate easier comparison, applications are grouped according to their major contribution in one of these areas.

2.2.1 Game Applications

For many years, games competed based on their visual effects. Now, games must also compete in terms of the gameplay experience they offer (Spronck *et al.*, 2006). One way to enhance gameplay experience is to provide large numbers of virtual characters that the player can interact with, for example *Oblivion* (by Bethesda Softworks) and *The Sims* (by EA Software). Although sometimes quite complex, these characters can appear too similar to their archetype. Another method to generate large numbers of characters is to use crowd simulators, such as those used in *The Lord of the Rings*, or for a forest fire simulation (Cho *et al.*, 2008). These simulators rely on giving characters simple behaviour and some fixed traits to present the appearance of diversity. The characters generated are often too simplistic to support player interaction.

Introduction of emotions into games has been seen as a potentially useful approach to enhance gameplay. Some middleware products have been developed to allow game designers to put emotions in their characters using the cognitive appraisal model (such as Sollenberger & Singh (2009)). However, much work in putting emotions into games is directed towards generating emotional responses in players (Freeman, 2004; French, 2007), or the graphical expression of emotions (e.g. Rehm & André, 2005), rather than enabling characters themselves to have and use emotions for decision-making. Characters developed often lack the social skill necessary for autonomous characters (particularly in role playing games, RPGs), so characters cannot become deeply involved in group tasks (Prada & Paiva, 2005). The future for games is likely to lie in creating more engaging games for the adult population that are not simply shooting or driving games (French, 2007).

There are roughly two ways to approach implementation of game AI. The first is the reductionist approach that reduces the number of entity types, but has a large number of instances of each type (Russell, 2008). This approach tends to homogenise the characters (Russell, 2008). The reductionist approach supports emergent gameplay, which gives a strong suggestion of open-endedness to players, leading them to believe

that they could continue to play and yet still encounter new ideas (Russell, 2008). The second approach is the constructivist technique where there are many different entities, but not many instances of each type (Russell, 2008). This approach promotes richness by using high levels of handcrafted work in individual scenes or characters to make memorable player experiences (Russell, 2008). However, this method has poor scalability and limits replayability (Russell, 2008). Although the player has a unique experience in a single playthrough, the experience is diminished on multiple playthroughs (Russell, 2008). Russell proposes the concept of situationist game AI that combines the reductionist and constructivist techniques and attempts to reconcile parallelism of action and conflicting situations (such as aiming a gun while opening a door) (Russell, 2008). The work presented so far appears preliminary and is primarily directed towards animating individual characters and groups of characters (Russell, 2008).

Across all these approaches, the actual techniques used to cognitively model characters are often "basic", including, finite state machines (FSMs), reactive behaviour rules, situation trees (Funge, 2000), scripts (Spronck *et al.*, 2006), and goal hierarchies (Adams, 2000). These techniques are often easy to understand and develop, but debugging or introducing changes to an existing system can be difficult. Characters often cannot adapt unless explicitly instructed, meaning that the characters cannot, by themselves, adapt behaviour in response to the skill level of the player or player preferences.

In the following sections we discuss applications and techniques with an emphasis on personality and, after this, adaptation. We then examine in detail Spronck *et al.*'s research group who aim to improve learning for strategy game characters.

2.2.1.1 Games with a Personality Emphasis

Early work on personality in computer games generally related to developing simple models of emotions, attitudes, moods and static personality traits for characters (e.g. Silva *et al.*, 1999; Wilson, 1999). This work recognised that these techniques were probably more useful to long term games (Silva *et al.*, 1999), rather than first person shooter (FPS) games (Wilson, 1999). When considering personality as part of the behavioural model, game AI developers generally seek simple models to provide the appearance of interesting and complex behaviour (e.g. Ulicny & Thalmann, 2002).

Some work relates to how to create avatars (the virtual representation of a player within a game) whose personality resembles the personality of the player themselves (e.g. Imbert & de Antonio, 2000). We did not find any mechanisms to semi-automatically create personalities in game characters that are distinct from other characters.

A key ingredient to providing distinct personalities is the creation of variety in the behaviour available to characters. There are different levels to this variety (Ulicny & Thalmann, 2002). At the bottom level, there is a single solution for a given task. At the next level, there is can be either a finite number of solutions or the solution can be composed of combinations of sub-solutions. At the highest level, solutions can be chosen from an infinite number of possible solutions (Ulicny & Thalmann, 2002). Ulicny & Thalmann (2002) have implemented a system that uses rules at the bottom level, hierarchical FSMs at the mid-level, and autonomous and scripted behaviour at the highest level.

Personality Types There are several examples of the reductionist approach to game AI in relation to personalities, in which a small number of personality types are developed, usually via handcrafting to suit the particular game. The personality types usually have entirely different behaviour, rather than tweaking personality parameters (e.g. Smith, 1999).

In da Silva Corrêa Pinto & Alvares (2005), five handcrafted and simple personality types are implemented for use in *Unreal Tournament* (by GT Interactive) with the aim of improving believability. They interpreted personality to relate to a character's motivations and goals, and how it acts to achieve its goals (da Silva Corrêa Pinto & Alvares, 2005). They took a working personality and obtained the desired personality by hand tuning global parameters and goal strengths, or adding a new module (da Silva Corrêa Pinto & Alvares, 2005). The authors believed that the number of concurrent actions able to be performed in their approach was not sufficient to be applicable to commercial games (da Silva Corrêa Pinto & Alvares, 2005). The entire model is very reliant on the domain's physical world, and consequently the developed characters have limited reusability. The personality types developed were static and stereotypical, did not use learning, and did not utilise different personas for different mood or emotions (da Silva Corrêa Pinto & Alvares, 2005).

Another example of creating personality types can be found in Ellinger (2008), who describes how to develop *archetypes* of personality that are instantly recognisable due to their one-dimensional nature, for example, "the coward", "the defender". These personality archetypes are not meant to be unique. Indeed they allow the player to use their existing knowledge of social interactions to determine how the archetype behaves and therefore which tactics work best against each particular archetype (Ellinger, 2008). For example, the player learns that "the coward" runs away. According to Ellinger (2008), more subtle distinctions in characters can be expanded using storytelling and dialogue, but are usually unnecessary since players fill in subtle behaviour themselves. Archetypes appear best suited to games for novice players or games that are not played for extended periods of time. After prolonged periods of time, players will instantly recognise each archetype, implement the counter tactics and easily defeat the character, thus eliminating the challenge element of the game, and rendering the game uninteresting in the eyes of many players.

Façade The game *Façade* represents pioneering work in giving agents emotions that affect behaviour (Mateas & Stern, 2002). In this game there are two distinct characters who (according to the story) are on the point of separating from each other. The player interacts with characters via text based conversation and from this discussion can choose to encourage them to split up or make their marriage stronger. The personalities of the characters were thoroughly handcrafted meaning it would be unrealistic to implement in more than a handful of characters. However, the game represents a break from the standard game genres and indicates a possible future for social games.

The Sims Although the characters in *The Sims* (by Electronic Arts) appear to be very complex, most of the "smarts" are stored within objects in the environment. These objects tell an agent what animations to display when using the object (Doyle, 2002). The object also lets agents know how this particular object can change the agent's emotional or social state (Doyle, 2002). The characters are unable to learn (Clarke, 2005). Personality is only modelled in these characters to the extent that their hierarchy of needs and some simple traits are different from other characters.

2.2.1.2 Games with an Adaptation Emphasis

In most computer games, the technologies used to build characters usually "constrain them to a set of fixed behaviours which cannot evolve in time with the world in which they dwell" (Merrick & Maher, 2006, p.1). Although some designers may use learning during game development, it is unusual to have games where characters learn in the shipped product (Kirby, 2005). In the preface to the latest AI Game Programming Wisdom book (number 4), Rabin (2008) lists three reasons why learning is not being used extensively, despite years of interest in the subject:

- Agents in games do not usually live long enough to benefit from learning.
- Learning happens over time, so it is hard for players to perceive, therefore benefits are subjective and unclear.
- Learning requires time-expensive trial and error and tuning.

All of this leads to a high risk (that the learning will not be noticed or useful) and time investment with benefits that are difficult to quantify, so it is hard for developers to justify including learning (Rabin, 2008).

There are a number of learning techniques that the games industry has used or investigated. Sanchez-Crespo (2005) provides an overview of machine learning techniques as applied to games. We will investigate applications using reinforcement learning, since this is the most applicable to our research.

Reinforcement Learning RL techniques (see theory Section 2.1.4.2, page 38) are commonly used in both games applications and intelligent virtual agents. Compared to other techniques, reinforcement learning allows character behaviour to be explained more easily, which is highly desirable feature for games (both from the designer and the game player's perspectives). The creatures in the game *Black and White* and the dog in *Fable 2* (both from Lionhead Studios) are created using a modification of the BDI architecture and a degree of learning (Champandard, 2007; Evans, 2002). However, the learning provided for the characters is restricted to reinforcement learning using feedback only from the player (Evans, 2002), so the characters are unable to assess by themselves what they personally consider "good" and "bad". This places an additional burden on the player to act as the external trainer. Since a player can only teach a limited number of characters, the technique is restricted to a few characters.

Merrick & Maher (2006) use motivation and an ϵ-greedy exploration strategy for RL applied to create support characters for massively multi-player worlds. The term "motivation" appears to refer to the difference between observation and expectation, where expectation comes from learning by clustering similar events together (Merrick & Maher, 2006). Their method can allow a single agent model to develop different skills for different agents when they are in different environments (Merrick & Maher, 2006), i.e. developing a form of personality for the agents. Although they claim this adds a highly desirable feature, the outcome appears to be a side-effect of their implementation, and there is no analysis of whether the differences are sufficiently distinct to achieve individuality.

Trait-based personalities have been built using a learning technique in combination with handcrafting, in order to get the best results (Pisan, 2000). Explicit models are better for games, so that they are easier to debug (Pisan, 2000). In this system, the next action is decided based on current state and history or memory (Pisan, 2000). The world is non-deterministic and characters have a single optimal way to act within the world (Pisan, 2000). Despite this simplification, Pisan (2000) found that the behaviour of the character when engaged in discovering the single ideal method was very interesting; to the point that delaying convergence of selection policy could be seen as desirable to prolong this period of interesting behaviour.

Game developers perceive it to be risky to allow characters to adapt after shipping, since the characters may develop undesirable habits and change the gameplay significantly. A combination of both online and offline Q-learning (a type of RL) can allow for the creation of characters with the capacity to adapt their skills to a specific human opponent after their initial training (Andrade *et al.*, 2005). This process allows Q&A testing to be performed on the character prior to game shipping and is likely to reduce the perceived risk to game developers.

An example of the application of RL to strategy games is found in the work of Spronck *et al.* and this is described in the following section that focuses on their research group.

2.2.1.3 Focus on Research by Spronck *et al.*

Extensive work has been done by Spronck, Ponsen *et al.* on applying advanced reinforcement learning techniques to combat and real time strategy (RTS) game characters.

Their main contributions are dynamic scripting (Spronck *et al.*, 2006) and hierarchical reinforcement learning (Ponsen *et al.*, 2006a). They have also compared learning techniques for a simple problem within the RTS world (Ponsen *et al.*, 2006a), and investigated ways to improve set up (Ponsen *et al.*, 2007) and speed (Bakkes & Spronck, 2006) of the reinforcement learning problem.

One of their aims is to make adaptive enemies who adapt tactics to find optimal tactics depending on the ability of their human opponent (Spronck *et al.*, 2006). The characters should be able to be used against both beginners and experts (Spronck *et al.*, 2006). Another aim is to reduce the complexity of the game and therefore allow the characters to learn more effectively (Ponsen *et al.*, 2007), and for each character to optimise its learning selection policy (Ponsen *et al.*, 2006a). Characters should be "interesting" (Spronck *et al.*, 2006). This "interest" applies to creating characters that can be beaten rather than generating opponent tactics that are unusual or captivating to interact with.

The applications implemented were designed to test learning techniques intended for RTS (strategy) games with a single opponent (Ponsen *et al.*, 2007; Spronck *et al.*, 2006). Ponsen *et al.* (2006a) used a simpler test world that was fully observable with one worker, one enemy, and one goal to achieve. In another article, Bakkes & Spronck (2006) used three grid world tests with different obstacles in the grid to determine which method of speeding up reinforcement learning achieved more successful characters.

Learning Details The reward function depends on the domain being used for testing. In Ponsen *et al.* (2007), the reward function for a particular state depends on game score which is measured using both military and building points. In the simple test worlds described in Bakkes & Spronck (2006), success was determined by how close the agent gets to the top row of the grid world. Other than the goal square, all other non-neutral squares were negative, e.g. causing death or decreasing health (Bakkes & Spronck, 2006). Determining a suitable reward function when agents pursue multiple goals is difficult, as found in Ponsen *et al.* (2006a).

The agents learn domain-specific knowledge or rules about what can be done in the world (Spronck *et al.*, 2006). They label state-action pairs with reward values (Bakkes & Spronck, 2006). That is, observations (i.e. states) and action pairs are stored with an associated assessment of success/reward (Bakkes & Spronck, 2006). After a new

observation, the reward is updated using an average of past value and current reward values (Bakkes & Spronck, 2006). In some circumstances, not only is the reward for the state visited updated, but a penalty can be attached to other actions available that were not taken (redistribution of reward) (Ponsen *et al.*, 2006a). Having a table with state-action values is appropriate for small domains, but states grow exponentially as the domain grows (Ponsen *et al.*, 2006a). This explosion of the state-action space is a major reason why standard reinforcement learning may not be suited to games (Ponsen *et al.*, 2006b). Even in simple worlds there are many possible states (Ponsen *et al.*, 2006a).

Standard reinforcement learning has difficulties determining the balance between exploitation and exploration (Spronck *et al.*, 2006). Some RL methods require the agent to know what states it can transition to, due to a system model (Ponsen *et al.*, 2006a). For games, due to the non-deterministic nature of player input, it is not usually possible to know all the states and the transitions between them to develop the system model. To overcome this, the RL technique, Q-learning, may be appropriate because it does not need a model of the system and is online (Ponsen *et al.*, 2006a). However, this technique is less effective for tactical or strategic level learning, where reward can be delayed and the agent can not determine final reward until other actions have been taken (Spronck *et al.*, 2006).

Initialising Domain Knowledge In order for Spronck *et al.*'s dynamic scripting technique to function, a good knowledge base is needed (Ponsen *et al.*, 2007). Some processes to provide this knowledge include manual coding, semi-automatic methods (machine learning techniques where strong tactics are pulled out for implementation), and automatic transfer from offline learning (where examples are annotated with state transitions) (Ponsen *et al.*, 2007). Ponsen *et al.* (2007) discuss using an evolutionary algorithm to generate the domain knowledge which is then used by dynamic scripting. This process constrains the action state space to reduce complexity inherent in a large numbers of states (Ponsen *et al.*, 2007).

Accelerating Learning Bakkes & Spronck (2006) discuss a method to facilitate faster reinforcement learning, by providing the characters with a more informed decision process when entering a state that has not been encountered previously. Using large

numbers of trials to establish "decent" behaviour takes a long time and the search may not be able to locate desirable behaviour (Bakkes & Spronck, 2006). In the proposed method, if the agent finds a state it has not been to before, it calculates a similarity value to determine which states visited previously are most similar to the current one and then uses this to determine the initial reward for the current state (Bakkes & Spronck, 2006).

Dynamic Scripting This method is similar in many but not all respects to reinforcement learning (Spronck *et al.*, 2006). Dynamic scripting changes individual scripts themselves. A script is built up of goals from the database (Spronck *et al.*, 2006), it is similar to a 'plan' in BDI terminology (see Section 2.1.1.1, page 21). Dynamic scripting only works when the game already uses scripts (Ponsen *et al.*, 2007). The method does not allow different personalities within the same agent class (Spronck *et al.*, 2006). Agents can choose rules (similar to goals) randomly, but these have changeable weights, so that the agent is more likely to choose some rules above others (Spronck *et al.*, 2006). The total weight on all rules is constant. Therefore, if the weight on "rule A" increases, then the weight on all others decreases (Spronck *et al.*, 2006). It is a key feature of the work of Spronck *et al.* (2006) that all rules are updated at every time step (Spronck *et al.*, 2006). Their work demonstrated that dynamic scripting can lead to combat behaviour optimisation (Ponsen *et al.*, 2006a).

Hierarchical Reinforcement Learning This method is useful when the agent is required to optimise two or more goals at the same time. The developer manually designs the hierarchy of goals (Ponsen *et al.*, 2006a) and decomposes tasks into simple independent subtasks within the goal hierarchy (Ponsen *et al.*, 2006b). In their implementation, they examined a case with two sub-goals: "move away from enemy" and "move towards goal" (Ponsen *et al.*, 2006a). A sub-goal is triggered based on how close the agent is to achieving the other sub-goal (Ponsen *et al.*, 2006a). Once a sub-goal has been chosen, the agent can choose a direction to move (e.g. north, south, east, west). Reward is calculated using an equal weighting of the two goals based on the position the agent was in before the choice compared to the position it is in after the choice (Ponsen *et al.*, 2006a). Convergence cannot be guaranteed. Although this is normally undesirable, it could be considered desirable in computer games where the

human player opposing the AI character can change (Ponsen *et al.*, 2006a). Hierarchical reinforcement learning appears to work well for two competing tasks but when there are more goals, it would be more difficult to develop reward equations and the hierarchical decomposition.

Summary of Research by Spronck *et al.* The research by Spronck *et al.* focused on sophisticated learning techniques based on redistribution of reward to improve tactics in strategy games. They tested different techniques to initialise domain-dependent knowledge and accelerate learning. The research used a hierarchy of goals so that reward could be calculated when there were two goals for the agent to achieve simultaneously.

2.2.2 Intelligent Virtual Agent Applications

Intelligent virtual agents (IVAs), or embodied conversational agents, have been used in a vast variety of applications, such as:

- a tour guide (Lim *et al.*, 2005; Zheng *et al.*, 2005);
- psychological models of the effect of oblivious ostracism (Selvarajah & Richards, 2005);
- teaching autistic children social behaviour (Dautenhahn, 1999);
- teaching school children about bullying (Dias & Paiva, 2005);
- military simulations designed to teach soldiers how to deal with emotional civilians (Si *et al.*, 2005; Traum *et al.*, 2005);
- interactive animals (Blumberg *et al.*, 2002; Seif El-Nasr *et al.*, 1998);
- planning (André *et al.*, 1999);
- robots in mazes (Gadanho, 2002);
- presentation teams (André *et al.*, 2000);
- interactive drama (Theune *et al.*, 2004);
- leveraging group social dynamics (Prada & Paiva, 2005);
- logistics (Buczak *et al.*, 2005); and
- coordination of multiple robots (Yingying *et al.*, 2002);

In this section, we begin with applications that relate predominantly to personality, then consider applications using adaptation followed by those using somatic markers. We finish this section with a focus on work done by Blumberg *et al.*'s research group relating to developing characters that adapt and also have their own personalities.

2.2.2.1 IVAs with a Personality Emphasis

According to Ortony (2002), believable characters should have variability within consistency. To achieve this, characters need to be coherent at a global level, across different kinds of situations, and over quite long time periods (Ortony, 2002). Characters also need to exhibit "within-individual consistency and cross-individual consistency" (Ortony, 2002, p.191). Personality (or constraining principles) can provide this consistency and emotionality can provide variability (Ortony, 2002). Personality gives life to characters, not emotions (Lim *et al.*, 2005). Certainly, for social systems, personality is a requirement (Campos *et al.*, 2006). The personality given to characters must be consistent itself (Francis *et al.*, 2010), because personality is viewed as a driver of behaviour (Ortony, 2002).

The model of personality does not necessarily need to be highly complex, since it has been shown (André *et al.*, 2000) that useful results can still be obtained with simple models. For example, Theune *et al.* (2004) found that, even with their limited implementation, a large number of different possibilities were able to be generated. Further, social responses can be triggered in users even if the agents are not very sophisticated (Rousseau & Hayes-Roth, 1997). Rousseau & Hayes-Roth (1997) implemented a simple system to determine whether personality can be detected. They found that simple personalities were detectable, but personalities that depended on moods and attitudes were hard to determine when the scenarios were not very long (Rousseau & Hayes-Roth, 1997). Their short sessions also caused them to find that adaptive personalities and extreme personalities were not believable (Rousseau & Hayes-Roth, 1997).

Usages of Personality Even within psychological and cognitive science theories of personality, the definition of personality varies greatly. Within applications, the usage of personality and how to model and store personality also varies as illustrated in the following examples.

- Personality is the thresholds that cause an emotion to be triggered (Taylor, 1995).
- Personality is defined based on high-level goals, with multiple ways to achieve goals (Mateas, 1997).
- Personality can be the preferences and long-term goals given to each character (André *et al.*, 1999).

- Personality is based on a vector of six possible actively-pursued desires (Parunak *et al.*, 2006).
- Personality includes OCC (cognitive appraisal) based goals, standards and preferences (Johns & Silverman, 2001), where preferences relates to opinions of objects and other agents rather than action preferences.
- Hard-coded personality can include goals, emotional reaction rules, action tendencies (reactive actions), emotional thresholds and decay rates for each emotion. Where emotion reaction rules are domain-dependent and cognitive appraisal rules based on personality (Dias & Paiva, 2005).
- Personality is modelled using emotional monitoring, personality evaluation and behavioural transformation (i.e. capable of changing coping preferences based on past experience) (Francis *et al.*, 2010).
- Memories can be seen as part of personality, particularly in reference to emotional memory (which relates to events and episodes) compared to semantic memory (which relates to facts) (Lim *et al.*, 2005).

Influence of Personality Just as personality can be used and implemented in a number of ways, it can produce different influences on the character itself. In general, personality influences the reasoning process (Dias & Paiva, 2005). According to Lazarus, personality influences both appraisal and coping (Lazarus, 1991), where appraisal generates emotions and decides which coping strategy to use and coping is the actual method an agent uses to deal with an emotional event. Table 2.1 lists aspects some of the major applications have used personality to influence. These aspects are: primary appraisal, decision-making, reward calculation, and goals or desires. Unlike most other methods, Rousseau & Hayes-Roth (1998) use personality to give actions themselves a personality profile. For instance, one particular action is labelled as something that only "extroverts" would perform.

It appears to be relatively common for personality to influence how decisions are made. For example, in André *et al.* (1999), personality and emotions were used as filters to constrain the decision process when selecting and implementing the agent's behaviour. It is less common for personality to influence reward calculation or evaluation of "good" or "bad". However, since some theorists believe personality should influence reward, we provide two examples here to demonstrate possible methods. Yingying *et al.*

Aspect Influenced	Implemented/Proposed By
Primary Appraisal	Gratch & Marsella (2004); Silverman & Bharathy (2005).
Decision-making (Secondary Appraisal)	Johns & Silverman (2001) and Silverman & Bharathy (2005); André *et al.* (1999).
Reward Calculation	Johns & Silverman (2001); Yingying *et al.* (2002).
Goals or Desires	Parunak *et al.* (2005); Lim *et al.* (2005).

Table 2.1: Aspects Influenced by Personality in IVA Applications.

(2002) modelled personality to affect evaluation (and not decision-making directly) in an application intended to allow multiple robots to coordinate assignment tasks between themselves more efficiently. They used evaluation weights (defined in relation to the personality) to change the total reward a robot calculates for itself (Yingying *et al.*, 2002). By allowing different robots to have different rewards, they will search for different optimal solutions and this is expected to improve coordination (Yingying *et al.*, 2002). Johns & Silverman (2001) used trait-based personality to obtain a single utility value from multiple emotions. The expected reward, i.e. utility, is calculated by multiplying each personality factor by the relevant emotion values to get a single utility value which is then used to determine which plan to choose (Johns & Silverman, 2001).

Separation in Reasoning Processes In some applications, the reasoning process (as applied to appraisal or decision-making) is separated into two parts, a quick process and a more deliberative one. Theories suggest that the brain completes a quick response without appraisal and then subsequently performs the (emotion) appraisal and responds more 'rationally' (LeDoux, 1996). For example, Dias & Paiva (2005) use a top-level to appraise the instant reaction, and a subsequent level for more thorough planning. Greene *et al.* (2005) use Damasio's somatic marker hypothesis as a reflex layer. André *et al.* (1999) separate the reasoning systems of affect and behaviour.

Similarly in Gadanho (2003), when making a decision, an initial emotional conclusion is made, which may then be rejected by a cognitive conclusion (Gadanho, 2003). The justification for this method is due to the dual purposes of cognition and emotion: "the cognitive system can make more accurate predictions based on rules [of causality] while the emotion associations have less explanatory power but can make more extensive predictions and predict further ahead in time" (Gadanho, 2003, p.386).

Emotions, Mood and Personality Some models implement personality, mood and emotion (PME). In these models, emotions last for a short time, mood is defined as a more general emotion that lasts for a longer time period, and personality is stable and unchanging. For example, Wilson (2000) sees personality as a kind of long term emotion. In these PME models, "personality" is trait-based and often uses similar terminology to that applied to emotions, e.g. a happy personality versus the emotion happiness. Work by Henninger *et al.* (2003) links emotions directly to personality, so that when there is high arousal according to the agent's emotions, the agent will revert to a 'core personality' or behaviour that has already been shown to work; otherwise the agent will try less safe choices. Some models (Egges *et al.*, 2004; Strauss & Kipp, 2008) implement a "generic" model of personality that can be used as a toolkit for other applications. The model is only generic in the sense that it can be applied to any trait-based model of personality, but not an adaptive model of personality.

Dias & Paiva (2005) implement mood as the overall valence of the emotional state, which is then used to influence the intensity of emotions. The intensity of emotions decays over time, according to an exponential function (Dias & Paiva, 2005). To calculate the intensity of an emotion, I, that was created based on an emotion event (or appraisal), k, after a given time, t, use a decay of b and following equation (Dias & Paiva, 2005).

$$I(k,t) = I(k,t_0) \times e^{-b(t-t_0)} \tag{2.4}$$

Applications using Cognitive Appraisal Model Many IVA applications have used the cognitive appraisal model to simulate emotions that the agents "have". For example, many have used the OCC model: André *et al.* (1999), Egges *et al.* (2003, 2004), and Seif El-Nasr *et al.* (1999). Dias *et al.* (2005) implemented the OCC model and a Lazarus style coping mechanism in their application, FearNot!. A very in-depth

implementation of Lazarus' cognitive appraisal model of emotions is provided by Gratch & Marsella (2004). They use heuristics to establish fixed preferences and then, in decision-making, to choose the most preferred coping strategy (Gratch & Marsella, 2004). This work emphasises the realistic generation of character emotions for small numbers of characters.

Personality Types Developed In many applications, a fixed number of hand-crafted, static personality types are developed (e.g. Dias & Paiva, 2005; Lim *et al.*, 2005; Rousseau & Hayes-Roth, 1997; Rousseau & Hayes-Roth, 1998; Yoon *et al.*, 2000). Personality can be hard-coded to make the character "interesting" (Rousseau & Hayes-Roth, 1997), or tailored by the designer using trait-based approaches so that each character type has its own way to exhibit behaviour (André *et al.*, 1999). The most common implementations of personality theories are trait-based theories relying on fixed personalities, for example Ball & Breese (2000); Jan & Traum (2005); Wilson (2000). Trait-based, hard-coded models of personality have been used to recreate fixed, stable personalities for characters based on past real-world leaders (Silverman & Bharathy, 2005). Bevacqua *et al.* (2008) used different emotional styles (similarly to personality traits) so that an agent who listens can choose statements that match to its emotional style and the apparent emotional state of the user. Some models, attempt to match the personality of the character to the personality of the user (e.g. Moon & Nass, 1996; Scheutz & Römmer, 2001). One reason for this is that it has been found that users want agents they must interact with, such as conversational agents, to become more like the user with time (Moon & Nass, 1996).

Situation-dependent Applications According to Mateas (1997), behaviour should be context-aware, but should be written for each individual character with their specific conditions. Some systems use fixed trait-based personality archetypes, but allow the archetype expressed to vary depending on the situation the character is in. For example, Rousseau & Hayes-Roth (1998) combined trait-based approaches with situated behaviour to allow traits to vary (according to probability distributions) to different degrees depending on the situation. In this way the designer can create an agent that is friendly only to people it likes (Rousseau & Hayes-Roth, 1997; Rousseau & Hayes-Roth, 1998).

Campos *et al.* (2006) also implement an entirely hard-coded, situated, trait-based static personality. In their system, behaviour is a function of the situation, the personality and a level of error (Campos *et al.*, 2006). The situated personality affects behaviour only, and behaviour does not affect personality (Campos *et al.*, 2006), so although inspired by Bandura, it is not a full implementation of the social learning theory (see Section 2.1.2.2, page 34).

In Satoh (2008), a museum guide agent senses its current context and uses this to tell a visitor pertinent information. However, in this simulation, context only relates to location, the character itself does not behave differently in different contexts, it simply provides different tourist information (Satoh, 2008).

Explaining Agent Behaviour For characters to be believable, it can be important (particularly for interactive dramas) for characters to explain their behaviour so that users can understand why a character chose a particular action (e.g Scheutz & Römmer, 2001). If the user understands what is going on in the mind of the character and its intentions, then character behaviour is more plausible (Wallis, 2005) and users tend to feel more comfortable (Yoon *et al.*, 2000). Scheutz & Römmer (2001) implemented autonomous agents who act on the behalf of the user when the user is absent from the virtual world. The agent's actions while the user is away are explained in an entertaining story (Scheutz & Römmer, 2001). Similarly, Theune *et al.* (2004) use a narrator to explain the actions of the characters so that the user can understand the character's motivations.

2.2.2.2 IVAs with an Adaptation Emphasis

Adaptation can be used to develop the initial personality of a character and allow it to expand or change. Learning or adaptation means characters can be interesting, even after long periods of interaction with them (Blumberg *et al.*, 2002). As with game characters, simple agents are easy to develop, however they can become predictable and brittle (Francis *et al.*, 2010). More complicated agents are more flexible but harder to develop (Francis *et al.*, 2010). Although adding adaptation to agents makes agents more convincing but they can become less controllable (Francis *et al.*, 2010).

A number of different learning techniques are used in the IVA domain. For example, Sanchez *et al.* (2004) use a combination of evolutionary learning, RL learning and

bottom-up intelligence. Seif El-Nasr *et al.* (1999) use RL learning for learning about events and Pavlovian conditioning for learning about objects. The complexity of their learning system is due to the need to address the more complex, non-deterministic input that is obtained from the user (Seif El-Nasr *et al.*, 1999).

Improving Reinforcement learning There is substantial work on methods to improve reinforcement learning. Reinforcement learning can be slow and needs to have some basic behaviour described first (Gadanho & Hallam, 1998). Driessens & Džeroski (2004) discuss how to improve a selection policy for applications where the rewards are sparse. Matignon *et al.* (2006) investigates how to improve convergence of RL techniques.

Role of Emotions in Learning Learning is not automatically linked to emotions. According to Gadanho & Hallam (1998) there are three ways that emotions can be integrated into the reinforcement learning process. Emotion can generate reinforcement reward values, emotion can determine the current state or emotion can trigger state transitions (for FSMs) (Gadanho & Hallam, 1998). For example, emotion values can be modelled to give expected utility (Bozinovski, 2002; Silverman & Bharathy, 2005). When this approach is taken, the problem of how to determine utility (or reward) for reinforcement techniques is solved, as long as emotion is implemented in a suitably complex manner. However, this is not always the case, because "computers do not automatically have valence attached to everything they learn; some mechanism must determine if the item is good or bad" (Picard, 1997, p.223). Often, reward is calculated based on feedback from the user, for example Francis *et al.* (2010); Seif El-Nasr *et al.* (1999); Velásquez (1998).

Emotions and the Adaptation Loop The use of emotion in decision-making and the adaptation loop is illustrated by the work of Ahn & Picard (2006), in which they aim to increase efficiency of learning and decision-making. The work is applied to practical problems where the goal state is obvious, such as gambling and maze-finding tasks (Ahn & Picard, 2006). The goal given to each agent is to maximise positive emotions and minimise negative ones (Ahn & Picard, 2006). Agents learn appropriate probability values for state transition functions (Ahn & Picard, 2006). Both long term

and short term achievement goals are considered, so that the agent may do a task that seems "bad" now, but will lead to greater reward (Ahn & Picard, 2006). The execution loop for each agent at each time step is (Ahn & Picard, 2006):

1. make a decision;
2. implement it (i.e. update the cognitive state);
3. determine reward;
4. update affect (emotion);
5. update uncertainty;
6. update extrinsic decision value;
7. move to new affective state;
8. move to next time step.

In their evaluation of their work, Ahn & Picard (2006) show that the agents are able to learn relatively quickly and converge on the optimal solution. That is, all agents learn the single correct optimal path to the goal state.

Learning Animation Sequences The emphasis of work by Sanchez *et al.* (2004) is for agents to learn the correct animation to show when requested by the game system. The agent must select actions that can achieve the requested task and construct a plan with the correct and minimal sequence of steps to achieve its goals (Sanchez *et al.*, 2004). The system is deterministic, so actions always have the same reward consequences (Sanchez *et al.*, 2004), which makes learning easier for the agents. Convergence of behaviour is not guaranteed (Sanchez *et al.*, 2004). Due to the way their system builds up an agent's selection strategy, the agents can develop slightly different behavioural modules so that each agent does not act exactly as its neighbours do, i.e. a form of personality is generated in the animations they present (Sanchez *et al.*, 2004).

Anticipation and Chromosomes Bozinovski (2002) uses anticipatory learning systems (originally designed to solve how to assign credit using a neural network) based on Dungeons and Dragons. This theoretical work applies a physics like view of personality using potential field, flow and tension (Bozinovski, 2003). Input personality is two traits (curiosity and patience) and a set of handcrafted "chromosomes" (Bozinovski, 2003). The chromosomes indicate to the agent which of the 20 situations (locations on a map) are "good", "bad" or "neutral", which affects their current emotional state.

For the neutral situations, the characters learn which behaviours allow them to move towards the "good" situations, i.e. they learn the selection policy (Bozinovski, 2003). Initial behaviour is based on the curiosity constant, whereas final behaviour is the learned behaviour (Bozinovski, 2003). Although situated behaviours are developed, all characters with the same starting personality have exactly the same behaviours due to the deterministic environments used for testing.

2.2.2.3 IVAs with a Somatic Marker Emphasis

Damasio's somatic marker hypothesis (Damasio, 1994) (see Section 2.1.3, page 36) has been implemented in a limited number of applications using intelligent virtual agents. The hypothesis has no true explanatory power. This means it cannot explain why a choice is "good", it simply attaches a positive or negative connotation with choices available to the agent in the decision-making process (Gadanho, 2003). Being able to feel "good" or "bad" does not "merely affect the agent's ability to learn, but helps it prioritise and choose among all its actions - learning, planning, decision-making, and more" (Picard, 1997, p.223). We now consider some applications that claim to be inspired by Damasio's work, but do not fully implement the hypothesis. Then we examine the body of work by two research groups who have used the somatic marker hypothesis in their applications.

Inspired By Somatic Marker Hypothesis McCauley (1999) was inspired by Damasio in their work based on Pandemonium Theory and applied to Wumpus work. Although inspired by Damasio, the model of emotions in Velásquez (1998), used for a robot exploring the physical world, does not extensively rely on Damasio's techniques for decision-making. The robot has a temperament (based on threshold levels) and learns emotions based on feedback from user (Velásquez, 1998). The robot has simple plans to choose from and more than one action can be performed at the same time (Velásquez, 1998).

The work of Ventura & Pinto-Ferreira (1999) claims to use somatic markers, but their implementation seems more akin to Pavlovian conditioning than somatic markers for decision-making. The system links images to a body state at the time the image occurred (Ventura & Pinto-Ferreira, 1999). This seems to be the wrong way around according to the hypothesis. In the somatic marker hypothesis, a particular body state

provides the individual with images (somatic markers) related to each possible choice to decide what to do. In the method of Ventura & Pinto-Ferreira (1999), the image triggers a body state (causing the face to change its expression), and then the agent decides whether the image was good or bad, based on some internal process.

Logistics and Military Applications The research group comprising Buczak, Greene *et al.* use somatic markers for agents in a logistics application and military simulations (Buczak *et al.*, 2005; Greene *et al.*, 2005). In their work, somatic markers are only used for reflex actions (Greene *et al.*, 2005). Their implementations are based on the military OODA (Observe, Orient, Decide, Act) loop and allows adaptation of reflexes at any stage in the loop (Greene *et al.*, 2005). The reinforcement learning process changes the reflex itself (Buczak *et al.*, 2005), and not the preferences for choosing to execute the reflex.

Their system is reactive; that is, a plan or reflex is only implemented if there is a stimulus (event) (Buczak *et al.*, 2005). If the agent has seen a stimulus before it will implement the previously learnt reflex; if not, it will attempt a new reflex (Buczak *et al.*, 2005). This adaptation occurs by following a series of steps:

1. When the agent takes an action (reflex), the agent predicts the result and creates an expectation object (Greene *et al.*, 2005).

2. The agent waits to see if it can match this expectation to an actual observation (Greene *et al.*, 2005).

3. If the agent does not find a match in time, then it assumes it has not met any of its expectations at all (Greene *et al.*, 2005).

4. If the agent matches an expectation to an observation, it compares that observation to the expectation and its reward is based on expected environment state to actual state (Greene *et al.*, 2005).

5. If the result is different from what is observed, then it may update the selection policy based on the summation of its reinforcement value over time (Buczak *et al.*, 2005).

When performance decreases, the well-being value of the agent decreases (moves away from ideals), and this decrease triggers the agent to find a better solution (Greene *et al.*, 2005). Exploration of new actions is also related to well-being (Buczak *et al.*, 2005). When well-being is low, the agent will explore more (Greene *et al.*, 2005).

Maze Finding Robots Gadanho's work implements the somatic marker hypothesis using a biologically based hormone system that alters the 'body' of the robot (Gadanho & Hallam, 1998). Gadanho uses somatic markers because they aid decision-making, according to Damasio. The application domain is the task of getting robots through a maze (Gadanho, 2003), a task that where the goal state is clearly defined based on a single dimension. The system uses only small number of emotions, since others are probably too sophisticated or irrelevant for the domain (Gadanho & Hallam, 1998). For example, love and hate are relevant in a social setting, but unlikely to suit a robot traversing a maze (Gadanho & Hallam, 1998). The cognitive and emotion models are entirely separate (Gadanho, 2003), similar to the separation of reasoning processes described in Section 2.2.2.1 (page 53). An initial decision is made based on an emotional decision, i.e. based on somatic markers. This initial decision may then be rejected by a cognitive decision (Gadanho, 2003). The emotion model is constructed from recent emotional history (Gadanho & Hallam, 2001). Emotions colour perceptions and are used for state transitions as well as utility (Gadanho & Hallam, 2001). Learning convergence is not guaranteed (Gadanho, 2003). Primitive behaviour is hard-coded as a base for learning (Gadanho, 2003). The primitive actions used were: avoid obstacles, seek light and wall follow (Gadanho & Hallam, 2001). This approach allows prediction of future outcomes of certain scenarios (Gadanho & Hallam, 1998).

2.2.2.4 Focus on Research by Blumberg *et al.*

Work by Blumberg *et al.* modelled both adaptation and personality in applications containing a small number of characters, such as a shepherd and dog (Isla *et al.*, 2001), puppies (Blumberg *et al.*, 2002) and three characters in a diner (Yoon *et al.*, 2000). The aim was to make virtual characters more compelling over extended periods of time by allowing them to learn (Blumberg *et al.*, 2002). Learning was also seen to assist the designer since "not every situation can be predicted at the character design stage" (Yoon *et al.*, 2000, p.365). The emphasis is on making characters learn movement

tasks, based on feedback from the user (Burke & Blumberg, 2002). The characters were required to be reactive and learn, in order to make "simple things simple and complex things possible" (Isla *et al.*, 2001, p.7). Interestingly, they found that some mistakes the characters made improved realism (Isla *et al.*, 2001).

Creature Kernel The main component of a character is its creature kernel which decides what the character does and how to do it (Yoon *et al.*, 2000). The kernel is made up of four systems: percept, motivation, behaviour and motor systems (Yoon *et al.*, 2000). The percept system handles how the character receives information from their world and the motor system implements chosen actions (Yoon *et al.*, 2000).

The behaviour system is a network of hierarchically connected units that can excite or inhibit each other and therefore govern the action selection process (Yoon *et al.*, 2000). The system triggers behaviour groups based on state, stimuli, interest, inhibitory gain and preference (Yoon *et al.*, 2000). The behaviour network can be modified by the agent and actions can be added and deleted as the agent learns (Yoon *et al.*, 2000).

The motivation system comprises drives and affect. Affect is emotions in a hierarchy (high-level affect is mood) that each have a valence (good/bad), stance (approach/avoid) and arousal (intensity) (Yoon *et al.*, 2000). Drives are also in a semi-hierarchical network with connections that are modifiable by the agent (Yoon *et al.*, 2000). The agent starts with species-specific drives such as curiosity, hunger, dislike of objects (Yoon *et al.*, 2000).

Choosing and Implementing Behaviour When deciding which action to implement, the action with the highest expected reward is chosen (Burke & Blumberg, 2002), i.e. a greedy policy. Only one top-level behaviour is active at a time (Blumberg & Galyean, 1997). A behaviour plan gives basic action commands and their importance to the motor controller system for the agent (Blumberg & Galyean, 1997). The motor controller implements startle actions (behaviour) first, then default ones, where startle actions can interrupt the current action (Isla *et al.*, 2001). Level of interest determines whether an action (behaviour) is interesting enough to implement (Blumberg & Galyean, 1997). There is a releasing mechanism which gives actions a value above which the action is triggered (Blumberg & Galyean, 1997). In Yoon *et al.* (2000), players can 'possess" characters which strongly encourages the behaviour system to allow

the player's requests to be executed, however, the characters can resist possession to the point that they leave the diner and the control of the player.

Learning Organisation, Concepts and Affective Tags Characters learn based on feedback from their own personal experiences, but also from observing other characters in the environment (Yoon *et al.*, 2000). Observational learning assumes the characters know where to focus their attention and what actions are interesting, i.e. exactly what should be noticed (Yoon *et al.*, 2000). In Yoon *et al.* (2000), the characters in the diner can learn via three methods: organisational learning, concept learning and affective tag formation.

Organisational learning modifies preferences on behaviour groups within the behaviour system and can add new behaviour or strategies (Yoon *et al.*, 2000). The preferences are linked to groups (not individual actions) based on expected reward, which is, in turn, calculated based on expected valence and stance which are calculated using inference learning about parent and children nodes within the network (Yoon *et al.*, 2000).

Concept learning relates to learning the features (from the percept system) which are associated with objects or events (Yoon *et al.*, 2000). All characters begin with the same concepts such as "animals are scary". The characters then refine the concepts as they explore the world, so they can learn "tigers are scary", "small, grey animals (mice) are not scary" (Yoon *et al.*, 2000).

Affective tags are updated based on motivational feedback and used when there are no other cues to prefer one way over another (Yoon *et al.*, 2000). They relate to individual objects and events, which can be general, such as do not like red, or more precise, such as do not like red umbrellas (Yoon *et al.*, 2000). Affective tags help the agent choose by eliminating actions related to objects or events with negative affective tags (Yoon *et al.*, 2000). Affective tags are based on somatic markers, but instead of linking the tag with the action choice (as in somatic markers), they link the tag to objects or events that may be involved in the action choice.

Animal Learning based on Reinforcement Learning In Blumberg *et al.* (2002), the model combines unsupervised RL with supervised animal training techniques to train a dog for typical dog tasks, e.g. sit. They use online learning and assume that

the agent gets immediate feedback from its actions (Blumberg *et al.*, 2002). Classical conditioning learning (Isla *et al.*, 2001) is used to teach "interesting" movements (Burke & Blumberg, 2002). The agent learns causality relationships which are a list of time-related cause and effect relationships that the agent has observed (Burke & Blumberg, 2002). A limitation of their model is that it biases the agents to learn immediate consequences rather than extended action sequences (Blumberg *et al.*, 2002). Agents store state-action pairs that are typically accompanied by a numeric value representing future expected reward or the benefit from doing that particular action in the associated state (Blumberg *et al.*, 2002). Action tuples include information on what to do, when, to what, and for how long (Blumberg *et al.*, 2002; Isla *et al.*, 2001). The agents are able to rank states in a hierarchy (a percept tree) by themselves during the simulation (Blumberg *et al.*, 2002). The animals can learn new states based on vocal input from users (Isla *et al.*, 2001) and these are placed within the hierarchy as the agent learns.

Personality Personality types can be initialised with different starting biases, and then allowing the character to learn new motor skills (Yoon *et al.*, 2000). In the diner implementation, personality is described using emotion-terms, such as "angry", "happy", "fearful" (Yoon *et al.*, 2000). The three characters in the diner application were each given their own creature kernel to govern behaviour, although most characters had similar kernels excepts for initial biases towards desires, learning rates and more (Yoon *et al.*, 2000). Characters are able to learn to like actions they would not normally like on their own based on feedback from a player in the world (Yoon *et al.*, 2000).

Summary of Research by Blumberg *et al.* Blumberg *et al.* used learning and personality for a small number of characters. Each character is designed with its own specific creature kernel with fixed personality characteristics. Characters can learn in a number of complex ways, via user feedback and via feedback according to their own drives and motivations. According to Picard (1997), in Blumberg's work, the effects are global, "biasing or predisposing [the character] to certain behaviours or actions, without determining these behaviours or actions"(Picard, 1997, p.217). Blumberg *et al.*'s systems appear very complex with many domain-dependencies and very reliant on the low-level percept and motor systems. The main emphasis of the research is a

small number of handcrafted characters who are believable, rather than large numbers of characters with different personalities.

2.3 Building Blocks: Theories to Be Used in this Thesis

In this literature survey we presented a number of theories from the broad area of research relevant to this thesis. Here, we summarise the main theories that underpin our model.

From Agent Research Our model uses a BDI paradigm (see Section 2.1.1.1, page 21), so as to provide an established mechanism for agents to reason about their goals and plans, as well as failure recovery. We use both agent research and emotions to underpin *soft goals* which represent goals that enable the agents to determine what "good" and "bad' means for them (see Section 2.1.1.2, page 24).

The cognitive appraisal model of emotions (see Section 2.1.1.3, page 27) can provide a complex domain-dependent appraisal of choices process to enable agents to determine what an event 'means' to them, in terms of which emotion to elicit and the intensity of that emotion. In the theory there are three types of appraisal: primary appraisal, secondary appraisal and reappraisal. Many implementations of appraisal concentrate on primary appraisal. However, the work presented in this thesis has an emphasis on personality, rather than emotions, so does not implement a full version of the cognitive appraisal model of emotions. In our model, we implement secondary appraisal as *appraisal of coping choices* or decision-making to determine which action to choose when more than one action is available. We implement reappraisal as *evaluation*, which is the process by which emotions are generated after actions have been completed. Coping is implemented as a constant activity that agents pursue to improve their overall wellbeing in the form of *soft goals*.

From Personality Theories Our model is based on cognitive-social theories of personality (see Section 2.1.2.2, page 32). We implement a combination of two types of learning, learning by response consequences using a reinforcing function and self-reinforcement. The reward received by characters is generated internally and is determined based on their own personal goals or motivations. The reward value depends

on their own behaviour and also depends on what has happened in the world. Due to this, the reward can be considered as partly self-reinforcement and partly learning by response consequences.

For our model, we wish to mimic the development of individual differences automatically, so that a simple character is able to adapt to its environment in order to gain suitable complexity. Work by Ortony and Lazarus (see Section 2.1.2.3, page 35) relating to how individuals differ has contributed to constructing the causes and the way in which behaviour is generated in our model. According to Caspi & Roberts (1999), there are a variety of methods to measure differences in personality (see Section 2.1.2.3, page 36). These methods are related to the testing-based research sub-questions that we proposed in the introductory chapter (see Section 1.2, page 14) as follows:

1. Differential Continuity. Research sub-question 3b (individuals obtained).
2. Absolute Continuity. Research sub-question 1a (behaviour over time), research sub-question 1c (reward over time).
3. Structural Continuity. Research sub-question 3a (comparing characters), and individuality for Research sub-questions 1d and 2b.
4. Ipsative or Person-centred Continuity. Research sub-question 2a (behaviour in different contexts)
5. Coherence. Research sub-question 1b (learning specific, functional goals, confirming continuity between soft goals and behaviour).

From Somatic Marker Hypothesis We use Damasio's somatic marker hypothesis (see Section 2.1.3, page 36) to provide preferences for actions and the decision-making process. In the hypothesis, somatic markers are part of a physical body. We will not attempt to represent this physical body, and represent somatic markers as stored preferences. We use the hypothesis to dictate how our agents make decisions between actions. That is, actions are grouped according to their somatic marker preference into desirable and non-desirable actions, rather than other methods of action selection that are based purely on a probability function according to the exact preference. All actions that are grouped together can then be considered equally, ignoring their ranking within that group. These somatic marker preferences are inherently context-aware, so that the characters will make decisions based on past experience in that particular context. The

somatic marker values are adapted using the character's personal reward value based on their personal goals.

From Adaptation Theories The aim of this thesis is to develop a model of personality that allows characters to become individuals without handcrafting all behaviour. This thesis does not aim to make any new contributions to the field of adaptation and machine learning. We use simple techniques to reduce complexity in this aspect of our model. Hence, we use a process similar to reinforcement techniques, except the reinforcement value comes from internal goals, rather than an external trainer.

2.4 Summary of Literature Survey

In this chapter we presented theories and applications relevant to this thesis. While this body of past work has inspired our research and provides a basis for our model, there are perceived gaps in the past work. Our work is expected to be useful to automatically generate background or support characters. It is believed that the user will have many interactions with characters of the same "type" (or archetype), and yet each instance (i.e. character) of a type needs to be distinctly different from others of the same "type". In this way the appearance of diversity in the environment is improved, the player is constantly exposed to new characters none of whom is exactly the same as another character.

Giving characters personality will enable them to become more interesting since they will appear different from other characters based on the behaviour they choose and the way they act within the world. People do not act the same way in every situation they are faced with. Previous applications use trait-based, static, personality theories for their characters. This means that, to provide characters whose behaviour (and therefore personality) depends on the situation, each situation has to be handcrafted by the designer, and so the designer needs to predict all the situations in which the character may find itself. In order to reduce the level of handcrafting required by the designer, personality development theories, such as cognitive-social theories can be used.

Current applications for games and intelligent virtual agents (IVAs) do not allow character personality to adapt and be context-aware without extensive handcrafting

and are all based on trait-based personality approaches. If the characters can continue to adapt, then the characters will become more engaging over longer periods of time. Other applications using adaptation do so primarily so that characters can learn functional tasks where the goal is clearly defined or they can learn based on extensive feedback from the user. These processes are suitable for simple environments (with a clear goal) or for a small number of characters (so that users do not have to explicitly teach large numbers of characters how to behave).

The somatic marker hypothesis allows characters to make quick decisions based on their past experiences and context. Somatic markers have not been previously implemented alongside BDI approaches or explicitly linked to personality. Other implementations of the somatic marker hypothesis have used it to allow characters to make better decisions in functional applications or to improve the way a character uses its emotions. In addition to this usage, we use the somatic marker hypothesis to represent part of a character's personality. This is because learnt somatic markers guide a character's decisions, which in turn determines behaviour, the visible aspect of personality.

Now that we have established the background and grounding for our work, we are able to introduce our model of agent personality development.