Using Adaptation and Goal Context to Automatically Generate Individual Personalities for Virtual Characters

A thesis submitted in fulfilment of the requirements for the degree of
Master of Computer Science by Research

Jennifer Sandercock
(BEng(Hons)/BSc)

School of Computer Science and Information Technology
College of Science, Engineering and Health
RMIT University
August 2009
Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the context of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

--------------------------------------------------

Jennifer Sandercock
14th August 2009
I dedicate this thesis to:

Jonathan Gratch and Stacy Marsella whose work inspired me to pursue this research area.

The “Duval de L’Epinoy” by Maurice-Quentin de La Tour for keeping me smiling.

My agents who kept me on my toes and guessing.

“For there is nothing either good or bad but thinking makes it so.”

Shakespeare, Hamlet, Act 2, Scene 2.
Acknowledgements

Firstly, I would like to thank Alex Holkner for implementing the GUI-side of my game implementation and for assisting in the brainstorming process when deciding possible game designs.

I would like to thank Kaye Marion and Andrew Buelke for their generous help with clustering and statistics.

For assistance with JACK programming, I would like to thank Ralph Rönquist and the JACK support group.

Thanks go to Ana Paiva and João Dias of GAIPS at IST in Lisbon, Portugal for help in making the model more understandable.

Massive thanks goes to my supervisors, Vic Ciesielski and James Harland, who stepped up to save me from certain devastation.

Thanks go to my mother, Ellen Stoddart, for helping edit my thesis and for believing in me.

On a personal note I would like thank my friends who helped me keep going, particularly: Rebecca Zurrer, Neil Kirby, David Graham, Tina Smart and Wen-Ching Un. Thanks go, as well, to all my other friends who listened when I needed it. To my companions at university, for helping me stay relatively sane, thank you: Simon Duff, Antony Iorio and David Poutakidis.
Contents

List of Figures \textsuperscript{xv}

List of Tables \textsuperscript{xvii}

Glossary \textsuperscript{xix}

1 Introduction \textsuperscript{3}

1.1 Motivation \textsuperscript{4}

1.1.1 Motivating Vignette \textsuperscript{6}

1.1.2 Motivation from Psychology and Personality Theories \textsuperscript{9}

1.1.3 Motivation from Virtual Agents Domain \textsuperscript{11}

1.1.4 Motivation from Games Domain \textsuperscript{12}

1.2 Research Questions \textsuperscript{14}

1.3 Thesis Overview \textsuperscript{17}

2 Literature Survey \textsuperscript{19}

2.1 Theories \textsuperscript{20}

2.1.1 Agent Theories \textsuperscript{21}

2.1.1.1 Beliefs, Desires, Intentions (BDI) Agents \textsuperscript{21}

2.1.1.2 Goal Types and Motivations \textsuperscript{24}

2.1.1.3 Cognitive Appraisal Model \textsuperscript{27}

2.1.1.4 Measurement Techniques for Believability \textsuperscript{28}

2.1.2 Personality Theories \textsuperscript{31}

2.1.2.1 Trait-based Personality Theories \textsuperscript{32}

2.1.2.2 Cognitive-Social Theories \textsuperscript{32}

2.1.2.3 Individual Differences \textsuperscript{35}
## CONTENTS

2.1.3 Somatic Marker Hypothesis ........................................ 36
2.1.4 Adaptation Theories .................................................. 37
   2.1.4.1 Aspects that can be Learned .......................... 38
   2.1.4.2 Reinforcement Learning .............................. 38

2.2 Applications ............................................................. 40
   2.2.1 Game Applications .............................................. 41
      2.2.1.1 Games with a Personality Emphasis ............. 42
      2.2.1.2 Games with an Adaptation Emphasis .......... 43
      2.2.1.3 Focus on Research by Spronck et al. ....... 46
   2.2.2 Intelligent Virtual Agent Applications .................. 50
      2.2.2.1 IVAs with a Personality Emphasis ............ 51
      2.2.2.2 IVAs with an Adaptation Emphasis .......... 56
      2.2.2.3 IVAs with a Somatic Marker Emphasis ...... 59
      2.2.2.4 Focus on Research by Blumberg et al. ..... 61

2.3 Building Blocks: Theories to Be Used in this Thesis ............ 65

2.4 Summary of Literature Survey ..................................... 67

3 Agent Personality Development Model .................................. 69
   3.1 Model Components .................................................. 73
      3.1.1 Soft Goal Related Components ......................... 74
         3.1.1.1 Achievement Level of Soft Goals and Soft Goal Equations 75
         3.1.1.2 Context ............................................. 76
      3.1.2 Somatic Markers .............................................. 78
      3.1.3 Personality Template ...................................... 80
         3.1.3.1 Goal/Plan Hierarchy ............................ 81
         3.1.3.2 Soft Goal Personality .......................... 83
         3.1.3.3 Emotionality ..................................... 84
      3.1.4 Domain-dependent Knowledge .............................. 86

3.2 Adaptation Process ................................................... 89
   3.2.1 Appraisal of Coping Choices ................................. 91
   3.2.2 Evaluation ..................................................... 93
      3.2.2.1 Step 1 Individual Soft Goal Reward Calculation ... 94
      3.2.2.2 Step 2 Calculate Personal Reward ............ 95

x
## CONTENTS

5.1.2 Learning A Functional Soft Goal (Research Sub-question 1b) ........................................ 152
   5.1.2.1 Comparison to Expectations ........................................ 152

5.1.3 Behaviour in Different Contexts (Research Sub-question 2a) ..................................... 154
   5.1.3.1 Comparison to Expectations ........................................ 159

5.1.4 Chi-squared Test (Research Sub-question 3a) ................................................................. 160
   5.1.4.1 Comparison to Expectations ........................................ 160

5.1.5 Summary of Behaviour Results ......................................................................................... 160

5.2 Personal Reward ................................................................................................................. 160
   5.2.1 Reward Compared to Random Choice (Research Sub-question 1c) ................................. 162
      5.2.1.1 Comparison to Expectations ........................................ 162

5.2.2 Effect of Adaptation and Context on Reward (Research Sub-questions 1d and 2b) .......... 162
      5.2.2.1 Reward Across Modes .............................................. 162
      5.2.2.2 Reward Across Cases .............................................. 163
      5.2.2.3 Summary of Personal Reward Results ......................... 164
      5.2.2.4 Comparison to Expectations ........................................ 164

5.3 Individuality ....................................................................................................................... 165
   5.3.1 Effect of Adaptation and Context on Individuality (Research Sub-questions 1d and 2b) ........................................ 167
      5.3.1.1 Individuality Across Modes ...................................... 167
      5.3.1.2 Individuality Across Cases ...................................... 168
      5.3.1.3 Summary ............................................................... 169
      5.3.1.4 Comparison to Expectations ........................................ 169

5.3.2 Individuality per Character (Research Sub-question 3b) .............................................. 169
      5.3.2.1 Comparison to Expectations ........................................ 170

5.4 Discussion .......................................................................................................................... 171

5.5 Extra Results: Domain-dependent Measures ................................................................. 173
   5.5.1 Happiness ............................................................................. 174
   5.5.2 Friendships ........................................................................... 175

5.6 Summary ............................................................................................................................ 176
6 Conclusion

6.1 Addressing the Research Questions

6.1.1 Research Question 1: Adaptation

6.1.2 Research Question 2: Context

6.1.3 Research Question 3: Individuality

6.1.4 Implications Arising from Research Questions

6.2 Potential for Future Research

6.2.1 Personality Templates and Input Parameters

6.2.1.1 Testing Adaptation Further

6.2.1.2 Designing Specific Characters

6.2.2 Model

6.2.3 Visualisation of Personality, Context and Friendship Networks

6.2.4 Qualitative Testing

6.2.5 The Personality Guessing Game

6.3 Implications for Games

6.4 Final Words

Appendices

A Behaviour Over Time

B Learning A Functional Soft Goal

C Domain-dependent Results

C.1 Happiness

C.2 Friendships

C.2.1 Case 1: Clear Preference Against One Activity

C.2.2 Case 2: Multiple Ways to Achieve Goals

C.2.3 Case 3: Conflicting Goals

C.2.4 Case 4: Complex Soft Goal Personality

C.2.5 Case 5: Different Soft Goal Personalities

References
List of Figures

1.1 Battlestar Galactica: Model Eight .................................................. 5
2.1 Generic goal/plan hierarchy .......................................................... 22
2.2 Standard BDI execution cycle ....................................................... 24
2.3 Reciprocal determinism in cognitive-social theories ....................... 33
3.1 Agent components and beliefs ....................................................... 74
3.2 Possible partial goal/plan hierarchy from the village domain. Part A ... 78
3.3 Possible partial goal/plan hierarchy from the village domain. Part B ... 79
3.4 Expanded version of the BDI execution loop ................................. 83
3.5 Adaptation process ................................................................. 83
3.6 The adaptation process with a generic goal/plan hierarchy ............... 85
3.7 Effect of adaptation processes on components and beliefs ............... 86
3.8 What makes an individual an individual in our model ..................... 100
4.1 Three characters from the game ................................................... 105
4.2 Implemented domain-dependent goal/plan hierarchy used for all characters 106
4.3 Screenshot from the implemented game ...................................... 107
4.4 Screenshot from the implemented game ...................................... 107
4.5 Screenshot from the implemented game without a player included .... 108
4.6 Flowchart to update attraction values the character has insulted someone 113
4.7 Flowchart to update attraction values when the character has been told an insult ................................................................. 114
4.8 Extract of sample reward output ............................................... 119
4.9 Extract of sample somatic marker output ................................... 120
4.10 Two examples of how characters can be the same ........................ 129
LIST OF FIGURES

4.11 Four examples of how characters can be different.......................... 130
5.1 Sample character graphs for Case 1 of behaviour based on the individual 148
5.2 Sample character graphs for Case 2 of behaviour based on the individual 148
5.3 Sample character graphs for Case 3 of behaviour based on the individual 149
5.4 Sample character graphs for Case 4 of behaviour based on the individual 150
5.5 Sample character graphs for Case 5 of behaviour based on the individual 150
5.6 Anna’s Behaviour when moving for Case 4........................................ 154
5.7 Bec’s sample character behaviour graphs for two contexts in Case 1......... 156
5.8 Chloe’s sample character behaviour graphs for two contexts in Case 2..... 157
5.9 Deb’s sample character behaviour graphs for two contexts in Case 3...... 158
5.10 Gina’s sample character behaviour graphs for two contexts in Case 4..... 159
A.1 Case 1 behaviour based on the individual......................................... 194
A.2 Case 2 behaviour based on the individual......................................... 195
A.3 Case 3 behaviour based on the individual......................................... 196
A.4 Case 4 behaviour based on the individual......................................... 197
A.5 Case 5 behaviour based on the individual......................................... 198
B.1 Anna’s Behaviour when moving for Case 4........................................ 200
B.2 Bec’s Behaviour when moving for Case 4......................................... 200
B.3 Chloe’s Behaviour when moving for Case 4....................................... 201
B.4 Deb’s Behaviour when moving for Case 4......................................... 201
B.5 Elle’s Behaviour when moving for Case 4......................................... 202
B.6 Fran’s Behaviour when moving for Case 4......................................... 202
B.7 Gina’s Behaviour when moving for Case 4......................................... 203
B.8 Heidi’s Behaviour when moving for Case 4........................................ 203
C.1 Happiness and reference reward for two example characters.................. 206
C.2 Friendship network and attraction levels for Case 1 ‘normal’.................. 208
C.3 Friendship network and attraction levels for Case 2 ‘normal’.................. 210
C.4 Friendship network and attraction levels for Case 3 ‘normal’.................. 212
C.5 Friendship network and attraction levels for Case 4 ‘normal’.................. 214
C.6 Friendship network and attraction levels for Case 5 ‘normal’.................. 216
# List of Tables

1.1 Criteria for success to be used to evaluate testing-based research sub-questions .............................................. [16]

2.1 Aspects Influenced by Personality in IVA Applications ............................... [53]

4.1 Calculation of friendship points for two characters. .................. [116]

4.2 Cases used with soft goal personality templates ........................... [136]

4.3 Expected results for criteria for success related to behaviour ............ [138]

5.1 Results for criteria for success based on behaviour .................... [160]

5.2 Average reward values for test Cases ......................................... [161]

5.3 Significant differences for reward .............................................. [161]

5.4 Individuality for test Cases ..................................................... [166]

5.5 Average number of characters who are different from the majority ... [171]

5.6 Summary of results of testing-based research sub-questions based on criteria for success .............................................. [177]
Glossary

**Action** An activity or a plan. A set of steps that the character can execute that are visible to a player.

**Activity** (See also top-level activity) A collection of plans and sub-goals that are used to achieve a high level plan that constitutes doing “something” within the domain. Once finished, success can be determined using evaluation. For instance, one activity could be “make something”. A plan within that activity could be “make bread”.

**Adaptation** The process by which behaviour changes over time. It is a simple form of learning based on experience. In our model this is done using a form of self-reinforcement via reinforcement learning, in particular the reinforcement comparison technique, see Section 3.2 (page 82).

**Adaptive** Can change behaviour over time. Uses adaptation.

**Agent** The reasoning part of a character (as compared to the visual aspects of a character). That is, the part that decides what to do and how to evaluate itself.

**Appraisal of (coping) choices** Used synonymously with decision-making, see Section 3.2.1 (page 87). The way that appraisal of choices is used within our model matches to secondary (not primary) appraisal in the cognitive appraisal model (Lazarus, 1991) (see literature survey Section 2.1.1.3 page 27). Reappraisal is implemented as evaluation.

**BDI** Beliefs, Desires, Intentions. Used in reference to the BDI paradigm that agents are embodied in their virtual world (see literature survey Section 2.1.1.1 page 21). Agents can hold beliefs or knowledge about their world and have desires of what they would like to achieve. An agent’s intentions are a list of the current plans it is using to achieve its desires. Intentions should be non-conflicting.

**Behaviour** Manner of acting; the observable actions and reactions of a person. Always considered over a specific time period. Behaviour is used as a measure of effectiveness of the model to test criteria for success (see Section 4.2.3.3 page 124). Measured in our implementation by counting the number of times characters choose different actions (activities and plans) over a fixed output time period.

**Beliefs** From the BDI paradigm (see literature survey Section 2.1.1.1 page 21), knowledge (facts and subjective opinions) that the agent has or stores about others and the environment.

**Case** A scenario used for testing. In each Case, the characters are given different soft goal personality templates. See Section 4.2.4.2 (page 135) for a listing of Cases.

**Character** The visual appearance combined with an agent, i.e. what a game player sees.
Conflicting goals  Goals that cannot both be achieved at the same time. For example, you cannot have your cake and eat it too.

Context  The perceived current situation of a character. Current level of achievement of soft goals that the agent is pursuing, regardless of importance, see Section 3.1.1.2 (page 75). This value is not based on the agent’s history, it is simply what the agent is achieving now. For each soft goal the agent is pursuing, the achievement level is converted to a single letter representing: high (close to achieving this goal), medium and low (this goal is not being achieved well currently). The letters are combined based on the alphabetisation of the soft goal name to create the context. For example, if the soft goals are “have friends” and “have money”, the context “LH” represents the state where the agent has hardly any friends and a lot of money.

Context-aware  Characters who’s behaviour depends on their perceived situation. In our model this is, behaviour based on knowledge of current context, in terms of the soft goals the agent is trying to achieve.

Coping  According to the cognitive appraisal model of emotions [Lazarus 1991] (see literature survey Section 2.1.1.3 page 27) coping is a mechanism that we engage in to improve our overall emotional wellbeing. Coping can be physical actions in the real world, such as running away when scared, or an emotional re-evaluation, such as realising there is no need to be scared in the first place. In our model, coping refers to the domain-dependent plans that the agents can use to act within the world in order to improve their overall wellbeing based on achieving their soft goals.

Criteria for Success  The minimal set of tests (shown in Table 1.1, page 16) that must be satisfied for the implemented model to be considered to have addressed the testing-based research sub-questions. The criteria test whether the characters generated by the model are adaptive, context-aware and individual. The criteria are measured based on the quantitative values of: behaviour, reward and individuality.

Domain-dependent Knowledge  Beliefs (facts and opinions) specific to the implemented domain. In the theoretical model, they are used to calculate achievement levels of soft goals, and therefore individual soft goal rewards, which leads to personal reward. In our implemented domain, the beliefs are opinions - happiness, attraction towards others and facts - attraction from others, location, insults said and told.

Emotion  Related to feelings. There are many types of emotion, such as happiness, anger, fear etc... In our model we use the term to relate to any or all of these types. In our implementation we use the term to refer to a happy/sad scale that represents how close the agent is to achieving all their soft goals.

Emotionality  A set of values that represent how the agent reacts to events, the thresholds above or below which they define “good” and “bad”, and other learning related parameters, see Section 3.1.3.3 (page 81). Used in a similar sense to Ortony [2002] (see literature survey Section 2.1.2.3 page 35).

Evaluation  The process by which personal reward is determined, see Section 3.2.2 (page 90). During this process the following are updated: achievement levels, context, somatic markers and emotion. This process is an implementation of reappraisal according to the cognitive appraisal
model of emotions (Lazarus [1991]) (see literature survey Section 2.1.1.3, page 27) and occurs after every activity has been completed.

**Execution of a plan** Plans are like simple functions, they are executed consecutively line by line. The execution of a plan is simply following the steps in the plan and ensuring that none of the individual steps fail.

**Facts** Beliefs that cannot be changed by an individual agent (i.e. different from opinions). It is a belief that is based on information given to the character by the environment or from other characters.

**Goal/Plan Hierarchy** From the BDI paradigm (see literature survey Section 2.1.1.1, page 21). In particular, see the generic figure for the goal/plan hierarchy, Figure 2.1 (page 22). The goal/plan hierarchy is a representation of hard goals and the plans that can be used to directly achieve these goals. The hierarchy begins with a hard goal placed at the top. Underneath this are a number of plans that can achieve this goal. Each of these plans can post a number of sub-goals that each must be achieved for the plan to succeed. This leads to a hierarchy, for example domain-dependent hierarchies see Figure 3.2 (page 78) and Figure 4.2 (page 106).

**Goals** Something that an agent wants to achieve or maintain. There are a number of different types of goals in the literature (see Section 2.1.1.2, page 24). In our model, we use hard goals and soft goals. Hard goals are implemented in the goal/plan hierarchy. Soft goals are part of an agent’s personality template in the form of an agent’s soft goal personality. Soft goals can be conflicting, whereas hard goals cannot. Agents are given no knowledge of how to achieve soft goals, but achievement of hard goals is explicit within the goal/plan hierarchy.

**Hard goals** Concrete goals within an explicit goal/plan hierarchy. The designer must explicitly state how an agent can achieve these goals. For example, to achieve the goal “make something”, an agent can choose a plan such as “make bread”.

**Importance** Used for soft goals, also known as weight. This is a number on a scale of [0, 1], where 1 represents a soft goal that the agent really wants to achieve, and 0 is one they do not care whether they achieve or not. It is part of soft goal personality, see Section 2.1.1.2, page 24.

**Individual** Different from others based on observable behaviour. A property a character can possess. In our model, an individual character is comprised of a number of components and beliefs. Primarily the components and beliefs include personality template, somatic markers and domain-dependent knowledge. The extent to which a character is individual is measured using individuality.

**Individual Soft Goal Reward** The reward for a single soft goal based on the soft goal achievement level and the distance to the ideal soft goal value from an agent’s soft goal personality, see Section 3.2.2.1 (page 91).

**Individuality** In the general sense, individuality is what makes each of us unique and different from other people. To test our model we needed a quantitative measure of individuality to compare characters to each other based on their patterns of behaviour over the entire running time. The quantitative measure of individuality is a count of the number of differences between characters based on whether the behaviour (action choices) are significantly different for the top-level activities (see Section 4.2.3.3, page 126). Individuality is used as a measure of
effectiveness of the model to test criteria for success. Individuality can relate to a specific character (number of characters that character is different from) or to an entire run (total number of characters different from each other).

Learning Feedback Loop Also known as the adaptation process. After completing an activity the agent evaluates the personal reward for the activity and then feeds this back into the point where the decision was made to do that activity, see Section 3.2 (page 82). In our model, feedback is based on personal reward and updates somatic marker preferences.

Measure of Effectiveness Observed data from testing that is used to determine whether criteria for success are satisfied. Three measures are used: behaviour (based on a count of the number of times characters chose actions), reward (based on personal reward calculations for characters), individuality (a count of differences between characters based on behaviour).

Mode A scenario used for testing. The different modes cause the characters to use random choice when making decisions (‘adaptation off’); or do not allow the characters to distinguish between contexts when they are learning (‘context off’); or using the full model (from Chapter 3) where characters use the methods specified (‘normal’). See Section 4.2.4.1 (page 134) for further explanation of modes used for testing.

Opinion A belief that has a value judgement attached. For example, a character can store “I like Anna a lot”. The values on opinions can be changed by the character, if so desired.

Past Experience A lookup table of preferences based on past rewards. See somatic markers.

Personal Reward Also known as self-reinforcement value. The agent’s personal evaluation of how “good” it thinks the last activity was. This represents how close the agent is to achieving all of its soft goals, with more importance placed on different goals according to the agent’s soft goal personality (see calculation step in Section 3.2.2.2, page 93). Reward is used as a measure of effectiveness of the model to test criteria for success.

Personality Personality is the set of observable characteristics that make an individual themselves. In our model, we restrict the term to relate to observable behaviour. A character’s final personality is a combination of their initial personality template as well as their learnt preferences, or somatic markers.

Personality Template In the general sense a personality template represents the basic genetic set-up of an individual, actual personality emerges through life experience. In our model, a personality template is made of three components: a domain-dependent goal/plan hierarchy, a soft goal personality and emotionality values. See Section 3.1.3 (page 78).

Plan A set of instructions or recipe that the agent can execute. A plan can result in observable actions in the virtual world or can change its beliefs.

Player A human participant in the game or simulation.

Preference Value See somatic marker preference.

Reference Reward A value that is representative of all past personal rewards. Used to determine whether a personal reward for a particular activity was “good” or “bad” compared to all other activities that have been executed. To see how the reference reward is used see Section 4.2.2.3 (page 94); to see how reference reward is updated see Section 3.2.2.3 (page 97).
Reinforcement Comparison Technique  A simple reinforcement learning technique (from Sutton & Barto (1998)) that compares the current reward received with all other rewards that the agent has received, using a reference reward, see literature survey Section 2.1.4.2 (page 35). The technique specifies how to update decision-making selection policy. In our model, it is used to update somatic markers, see Section 3.2.2.3 (page 94).

Reinforcement learning Learning that is based on maximising reward from an external agent based on trial and error, i.e. punishment and reward, see literature survey Section 2.1.4.2 (page 38). In our model, we use learning based on self-evaluation (personal reward), not an external agent, and use the reinforcement comparison technique to update selection policy (somatic markers).

Research Questions  For this thesis, the three research questions relate to developing, implementing and testing a model of personality that is adaptive, context-aware and individual. Introduced in Section 1.2 (page 14).

Research Sub-questions  Detailed questions that breakdown the research questions into smaller parts. The sub-questions are also divided into model-based (relating to the development of the personality model) and testing-based (relating to how to determine success of the implemented model). The testing-based sub-questions are considered answered when the criteria for success have been satisfied. Introduced in Section 1.2 (page 14).

Reward  A measure of merit of an activity, i.e. “good” or “bad”. In our model there are three types of reward: individual soft goal reward which is reward for a specific soft goal based on its achievement level; personal reward which is based on combining all individual soft goal rewards based on an agent’s soft goal personality template (used as a measure of effectiveness of the model); and reference reward which is a running average of all previous personal rewards.

Soft goal equations  The domain-dependent functions used to determine the achievement levels of the soft goals. For a description of how they are used generically see Section 3.1.1.1 (page 74); for the specific soft goal equations used in our implementation see Section 4.1.2.3 (page 115). For example, if the soft goal is “have friends”, the equation to determine the quantitative achievement level could be based on such beliefs as: number of people the agent likes, number of people who like the agent, or a combination of these beliefs.

Soft goal personality  The soft goals this agent is trying to achieve, the importance levels it places on the goals, and the ideal value of each goal, see Section 3.1.3.2 (page 80). Once set for an agent, this will not change. For example, in one domain the agents may be able to have the soft goals “not be hungry” and “have money”. One possible soft goal personality is that the agent places a high importance on “not be hungry” and a medium importance on “have money”. Importance is a number on a scale of [0, 1], where 1 represents a soft goal that the agent really wants to achieve, and 0 is one they do not care whether they achieve or not. For each of the soft goals that the agent is trying to achieve, an ideal or maximum value is specified. For example, one agent may consider “have money” achieved when they have $100,000, another may believe they need $1 million.

Soft goals  Soft goals are a set of potentially conflicting goals that the agent is attempting to achieve at every step, see Section 3.1.1 (page 73). Plans may contribute partially to achieving a number of different soft goals. Some examples of soft goals are: have friends, have money and not being hungry. Agents do not initially know how to achieve their soft goals, they must learn.
For example, if the soft goal is to “have friends”, the designer developed domain-dependent goal/plan hierarchy does not need to have a plan that can directly achieve this, i.e. there is no need for a plan called “make friends”. Plans such as “interact” or “give away something” may improve the achievement of the soft goal, but this can only be learnt based on feedback from trial and error.

**Sub-plan** Some plans require more hard goals to be posted to finish the plan. These goals will be handled by sub-plans. These are plans that are beneath the originating plan in the goal/plan hierarchy. For example, to achieve “make something” the goal “choose what to make” is achieved by implementing a sub-plan such as “make bread”, see Figure 3.2 (page 78).

**Top-level Activity** A plan that is very high up in the goal/plan hierarchy. It should be something that has a long enough duration that the agent’s domain-dependent beliefs will have changed and the agent can perform an evaluation of what has happened. For example, it would be difficult to perform an evaluation after a small step, such as choosing who to talk to. A suitable top-level activity would be a longer interaction, such as actually having an entire conversation with a character. In our implementation, the agents have three top-level activities: “move”, “insult” and “wait”, see Figure 4.2 (page 106).

**Weight** Used for soft goals, also known as importance. This is a number on a scale of [0, 1], where 1 represents a soft goal that the agent really wants to achieve, and 0 is one they do not care whether they achieve or not. It is part of soft goal personality, see Section 3.1.3.2 (page 80).
Abstract

Personality is a key component of characters that inhabit immersive virtual environments, such as games and virtual agent applications. In order to be distinguishable from other characters in the environment, each character should have its own personality in the form of different observable behaviour, not solely in its physical appearance or animation. Previous work in this field has mostly relied on time-consuming, handcrafted characters and static, trait-based approaches to personality. Our goal is a method to develop complex, individual personalities without handcrafting every behaviour. Unlike most implemented versions of personality theories, cognitive-social theories of personality address how personality is developed and adapts throughout childhood and over our lifetimes. Cognitive-social theories also emphasise the importance of situations in determining how we behave. From this basis, we believe that personality should be individual, adaptive, and based on context. Characters in current state-of-the-art games and virtual environments do not demonstrate all of these features without extensive handcrafting.

We propose a model where personality influences both decision-making and evaluation of reward. Characters use their past experiences in the form of simple somatic markers, or gut-instinct, to make decisions; and determine rewards based on their own personal goals, rather than via external feedback. We evaluated the model by implementation of a simple game and tested it using quantitative criteria, including a purpose-designed individuality measure. Results indicate that, although characters are given the same initial personality template, it is possible to develop different personalities (in the form of behaviour) based on their unique experiences in the environment and relationships with other characters. This work shows a way forward for more automated development of personalities that are individual, context-aware and adapt to users and the environment.
Chapter 1

Introduction

As virtual worlds become more complex and visually believable, there is an expectation that the characters that inhabit these worlds likewise evolve to match their world. Personality is a key component of believable characters (Mateas, 1997). Often personality is approached from an animation point of view (e.g. Stuart, 2007) to the exclusion of considering how personality affects behaviour or reasoning. Visual appearance of a character can give the false impression of an underlying difference between characters. However, over time, character differences will only be noticed if they result in differences to observable behaviour. Personality should affect the way a character evaluates its personal success, makes decisions and, as a direct result, how the character behaves in the world. Within the field of games and virtual agents, there are many implementations of personality. Usually these implementations require handcrafting each individual character in a way that is not suitable for large numbers of characters, or rely on a set of fixed archetypes where every character within the type is exactly the same. As a result, environments with large numbers of characters often develop characters that do not withstand close inspection. We aim to develop a method of automatically generating individual characters with their own unique personalities in order to create more believable characters in virtual environments with large numbers of interactive characters.

Personality should be unique for each character, i.e. individual. Cognitive-social theories of personality address how personality is developed throughout our lifetimes. According to these theories, two essential components of personality are that it should be able to adapt or change; and that situation or context is important in determining
1. INTRODUCTION

how we behave. That is, in real life, people react differently in different contexts based on their past experience and the underlying goals they are trying to achieve. The term context could relate to a physical location or other participants in the activity. We will be looking at the case where context relates to how close the character believes it is to achieving its goals. For example, one context could be that the character has achieved one of its goals, but not another. Over time people adapt their behaviour based on past experience of different contexts. People will adapt and change their own behaviour, and to some extent their personality, based on other people they interact with and changes in the environment. Characters within current state-of-the-art games and virtual agent environments do not demonstrate all of these features unless explicitly handcrafted to do so. We will be concentrating on observed personality as a sufficient requirement instead of “actual” personality. We do this because human users interacting with these characters can only distinguish characters based on their observed behaviour. From this basis of cognitive-social personality theories, we believe that personality should be: individual, adaptive, and based on goal context. We now present our motivation for this thesis followed by a description of the research questions that will be addressed within this thesis.

1.1 Motivation

This thesis addresses important issues about how to make game characters and virtual agents more realistic or believable by using cognitive-social personality theories. We begin motivation by discussing two examples. The first is from a television series and the second is a more concrete example that will be used throughout the thesis to explain concepts and the model. These examples represent the ideal goal of our work. After these examples have been presented, we discuss where the theories that motivate this work have come from, as well as why games and virtual agents are in need of the work presented here.

Our first example come from the new version of Battlestar Galactica (by the SciFi channel) which is about a future war between humans and Cylons. Cylons are similar to robots and have been built by humans according to the back story. Some Cylons have been built to look like humans externally. There are 12 different “models” of these human-looking Cylons. A model includes the physical appearance of the Cylon
1.1 Motivation

(a) Boomer  
(b) Athena  
(c) Number Eight  
(d) Copies of Model 8

Figure 1.1: Model Eight from *Battlestar Galactica* (produced by the SciFi channel): There are many copies of model Eight (also known as Sharon Valerii). Each copy is generated from the same template, and yet takes on its own unique personality.

and some personality characteristics. There are many Cylon copies built of each of the models, so that externally each of these Cylons can appear to be the same “person” at first glance, e.g. Figure ?? However, each individual Cylon, or copy, from the same model is able to build up their own memories based on their experience and this changes their future behaviour and eventual personality.

We look at the example of model number “Eight”, Sharon Valerii, see Figure ?? Within the series we are introduced to this character without realising she is a Cylon. In fact, she does not realise herself that she is a Cylon until later in the story. As the series progresses, we are introduced to other copies of number Eight. Each copy has her own experiences and after many episodes we are able to distinguish at least three different “Sharons”. The first Sharon we meet is nicknamed “Boomer”, Figure 1.1(a) Boomer has a romantic relationship with the chief human engineer but eventually sides with Cylons against the humans. Next is Athena (figure 1.1(b)), who has always known she was a Cylon, unlike Boomer. Athena fights on the side of the humans, is in a relationship with a human and has a child with him. There are also many copies who remain with the main Cylon attacking force. Mostly these copies are indistinguishable from each other. However, as the series progresses the “head” of the number Eights shows increasing interest in Athena, to the point that she downloads Athena’s memories.

\footnote{Figures a and c taken from \url{http://www.thescifiworld.net/interviews/grace_park_02.htm}  
Figure b from \url{http://www.imdb.com/media/rm4068644352/tt0407362}  
Figure d from \url{http://en.wikipedia.org/wiki/File:Battlestar_Galactica_1x12_Number_Eights.jpg}}
1. INTRODUCTION

and yet still remains different from Athena. This copy also heads a group of Cylons who want to help the humans rather than destroy them.

What is interesting about this example is that the unknown designers of the number Eight model set in place only one personality template. The different copies become distinguishable based on their own experience, with no extra input from the designers. The designers are able to generate a personality template for each model and then allow copies to become their own individuals over time, while retaining some behaviour similar to the other copies within that model number. This example is motivating because the designers handcraft only twelve Cylon models and yet each of the many Cylon copies can have its own individual personality. This ability to produce many different characters from a handful of templates represents the overall goal of this thesis, i.e. to be able to generate individual characters with personality without handcrafting all behaviour.

To motivate our work further, we describe an example of an ideal virtual environment that could be built using the techniques that are developed in this thesis. This example or vignette will be used throughout the thesis to explain concepts and the model.

1.1.1 Motivating Vignette

Imagine a virtual world populated with diverse characters who live in villages and who each have their own personality distinguishable from others. Usually in this kind of world, the personality templates that are used become visible to a game player after prolonged interaction with the world. For instance, if you have met the baker in one village and travel to another village, you will probably meet essentially the “same” baker. Imagine if this new baker still had many of the core traits of the other baker, but had a different personality, i.e was recognisably different. For example, the baker in the first village might sing while serving his customers, while the second might be sullen and grumpy towards his customers, but may make better quality bread. Finally, imagine that these differences are not handcrafted, but are automatically developed based on each character’s personal experience with other characters, including the player, and the world. Further, if something changes in the world, such as a stranger entering it, the characters’ personalities will change over time to reflect their new experiences. For
example, if the stranger is very hostile existing characters will change their behaviour around the stranger and perhaps with other characters as well.

This is the world we are attempting to achieve.

Let us look at this example in more detail, since we will be using it as a case study throughout the thesis, beginning by looking at how to build one village. In our goal model, a limited number of carefully crafted personality templates would be designed. A personality template includes information about how the character reacts to events, what their personal goals are and the actions or choices available to it within the world. These personality templates form the building blocks of all the characters. Characters that have the same initial personality template will not necessarily have the same resulting personality, similarly to identical twins and Cylons gaining their own experience and becoming different from their twin or, in the case of Cylons, their model number. Personality is generally considered to be a result of both nature and nurture. In our goal model, the personality template represents a character’s nature, whereas experiences in the environment represent a character’s nurture.

Using these personality templates, we generate a large number of characters for our initial village. Then we allow the characters to interact with each other and learn how to achieve their personal goals. This allows them to take on roles within the village, such as butcher, baker or candlestick maker. Characters learn which action they prefer to take and how they prefer to execute the actions. These preferences are different depending on the character’s perceived context or state of its current goals. For example, when a character is not achieving its goal to make friends, it may prefer to talk to someone new, whereas when it has many friends it may prefer to do something different. Characters with the same personality template will not be exactly the same. This is because of the different relationships developed and the different methods the characters have found to achieve the common goals. Now we have a village full of interesting characters who all have their own history and relationships with the other characters.

Next we use the same starting personality templates, but with a different random seed, for a second village. Again we allow the characters to interact with each other and develop their preferences and personalities. We now have a new village full of interesting and different characters, and yet these characters will be observably different from the characters in the original village. A human player who walks between the two villages
1. INTRODUCTION

will see some similarities, but will not be faced with “the baker” who is always the same in every village. For example, in one village the candlestick maker and the baker might be married, in another village they may never have met.

Villages will be able to generate a community personality of their own. For example, if all the characters have a goal to make sure everyone is not hungry, then one village might rely on giving away goods for free and then receiving other goods for free, whereas another village works only by the use of money or bartering. If a baker is trying to be generous and giving away bread, it will still need meat and candles. Unless the other characters cooperate and give it these items, it may become hungry or will not have light.

The villages will not be able to function well unless the roles of ‘butcher’, ‘baker’, and ‘candlestick maker’ are filled (there may be more roles in an actual game). However, who takes each of these roles is not defined by the personality template; it is discovered by the character as a suitable way of achieving their personal goals. For instance, the goal of making money could be achieved by any of these roles, but would probably not be successful unless the character specialises in only one of the roles. The model proposed in this thesis reflects the real world, where some personality “types” are more inclined towards certain professions or roles. However, personality is more than someone’s job. The way that a character executes their role and their overall behaviour informs the observer of their entire unique personality.

One issue encountered in the games industry, is that characters capable of learning can be considered risky because they may develop new behaviour that may offend the players or change the gameplay significantly. The benefit of this model is that since the personality template restricts the actions available to the character (and this does not change over time), the character’s actions cannot become entirely unpredictable. That is, characters cannot generate new actions, they can simply choose differently. What gives the diversity and appearance of a level of unpredictability is the choices the character makes and its preferences for those actions. An individual character, if watched, will eventually become predictable in their actions. However, if a player sees a character of the same personality type they will not be able to entirely predict the new character’s behaviour without watching it for some time. By using the model in this way, characters will not be able to generate “dangerous” or unconstrained behaviour that becomes unsuitable in a shipped product.
1.1 Motivation

1.1.2 Motivation from Psychology and Personality Theories

When considering the motivation to pursue this topic, we need to establish why personality should be an important aspect of any virtual character. Chuck Jones, a cartoonist for Warner Brothers, found that in creating believable characters, personality is the most important aspect: it is “the individual, the oddity, the peculiarity that counts” (Jones, 1994, p.14). Personality and emotions have been successfully used as “filters to constrain the decision process when selecting and instantiating the agent’s behaviours” (André et al., 1999). Personality can be considered as the engine that generates reactions and responses in a coherent, consistent and predictable manner (Ortony, 2002).

We believe that personality is visible in the observed behaviour of characters. Therefore, the action a character chooses out of many possibilities should be a reflection of the character’s personality. Differences in behaviour for the same person and for different people are due to a number of factors including emotionality (differences in emotional reactions to events), current state and interpretations of the world situation (Ortony, 2002). Or, according to another theory (Lazarus, 1994), people respond differently to similar events depending on their individual goal hierarchies and perceived current context of themselves and the world. That is, behaviour or responses are context-dependent. We believe that context should be based on the current level of achievement of goals. For example, if a character currently has a lot of money but no friends, it may achieve his goals better if it gives away food, rather than selling it. We will take these concepts into consideration when building our model for personality that generates behaviour.

Trait-based theories of personality are popular for virtual characters. However, trait-based theories assume that personality is static and unchanging, and offer no explanation of how personality is developed (Cervone & Pervin, 2008). Although this may be appropriate for many environments, it assumes that the designer is able to develop a suitably complex personality for every character in the environment and predict the situations the character will encounter. Cognitive-social theories of personality (for example Bandura’s social learning theory (Bandura, 1977)) address the issue of personality development. If we allow characters to develop their own personality according to cognitive-social theories, then we will be able to use a simple template to generate
1. INTRODUCTION

many different complex personalities. This process will hopefully reduce the burden on
the designer.

Cognitive-social theories believe in reciprocal determinism: that is, behaviour re-
results from the complex interaction of persons and the environment, rather than from
any single factor alone (Bandura, 1977). Hence, people are neither driven by inner
forces nor buffeted by environmental stimulus. The traditional view of behaviour in-
teraction is that a person’s behaviour is a function of the person and the environment.
However, people’s actions contribute to the overall environment, which will in turn
reflect behaviour in a reciprocal fashion (Bandura, 1977). In our village example, if
one character gives away food it may cause those around it to also give away food,
i.e. the environment has changed because of one character’s behaviour. Experiences
that a character generates through their own behaviour will affect what a person be-
comes. That is, if a person tries to bake bread (behaviour), and they are successful,
they may continue to do this (changing the person). Both the person and the environ-
ment will in turn affect subsequent behaviour (Bandura, 1977). There are three main
types of learning according to Bandura, by response consequences, through modeling,
and self-reinforcement (Bandura, 1977). People learn and adapt via these mechanisms
throughout their lives. Further, according to cognitive-social theories, behaviour is not
determined by global traits, behaviour depends on the situation that the person is in
(Cervone & Pervin, 2008).

An inspiration for this thesis comes from Damasio’s somatic marker hypothesis
(Damasio, 1994). According to this hypothesis, emotions in the form of “gut instinct”,
called somatic markers, guide the decisions we make. Without somatic markers it
would be a struggle to interact rationally with other people (Damasio, 1994). Somatic
markers are built up automatically throughout our lives, so that when faced with a
possible decision in a particular context, they guide us towards or away from certain
choices. After this (often unconscious) elimination process, we are able to make a more
studied analysis of choices available in order to decide which to choose.

Cognitive-social theories and the somatic marker hypothesis represent ways that
real humans interact with other humans. Somatic markers represent an aspect of per-
sonality, in that our gut instinct guides us towards certain choices in a way that is
different from how another person may be guided. Somatic markers are based on past
experience, and we will use cognitive-social learning theories to inform how our characters acquire these somatic markers. This acquisition process and using somatic markers for decision-making is based on the character’s context, to reflect that people choose different actions in different situations. In our thesis, we implement a combination of learning by self-reinforcement and by response consequences using a reinforcing function to build up actual somatic marker values and therefore influence behaviour and the environment.

From cognitive-social theories and the somatic marker hypothesis, we believe that personality is unique to each individual, it should adapt to the environment and should be context dependent. As we will show, these factors are not fully implemented in current computer games and virtual characters.

### 1.1.3 Motivation from Virtual Agents Domain

Virtual agents can be seen in a number of application areas, from military simulations to pedagogical environments to embodied conversational agents. Believability is one of the key goals for most research groups in this area. Believability will make the characters more engaging and users will have a more enjoyable experience (Johnson et al., 2000). Characters that have unrealistic behaviour are more noticeable and distract users from the virtual world (Johnson et al., 2000).

It has been thought for a long time that the use of emotions in virtual environments improves decision-making and believability of characters (Minsky, 1986). Over the past fifteen years, there has been increasing use of emotions to improve realism and believability of intelligent agents in the agent research field with much initial success. This research by others into how to implement emotions in virtual agents is important, but often does not fully address the issue of how to implement personality. Personality gives life to characters, not emotions (Lim et al., 2005). Some research has concentrated on how to animate characters with their own personality, i.e. change the visual appearance of behaviour. Our work concentrates on how to imbue the character’s decision-making and evaluation processes with their own personality without handcrafting every step. We believe this will contribute to character believability. We concentrate on implementing personality that is unique to each individual; adaptive over time; and reflects the context the character is in.
1. INTRODUCTION

In the virtual agents domain, most characters developed can generally be considered to be individuals. However, this is often because there are no other characters to compare against, or because the individuality has been handcrafted, for example, the Oz project \cite{Mateas1997}. The Oz project gave equal attention to believable characters and story or interactive drama. However, the characters were handcrafted to obtain the desired effects.

Most implementations of personality are based on trait-based theories, such as the popular Five Factor Model \cite{McCrae1992}, and are static with respect to time and contexts. However, personality does develop over time, particularly during childhood experiences or when the environment itself changes in a substantial and long term manner. For instance, a character that is constantly ignored would be expected to change its behaviour over time to reflect this. Over extended periods of time, characters will be more compelling if they appear to learn from experience \cite{Blumberg2002}. However, characters should adapt in a way that is consistent with their personality \cite{Mateas1997}. Characters such as Blumberg’s dog \cite{Isla2001} can adapt, but not in conjunction with its personality and only when taught explicitly by the user, i.e. not by itself. Static implementations of personality do not provide support for context-aware behaviour. In a static implementation, the number of starting personalities is the same as the final number of personalities. Since the characters cannot learn, their personalities will be fixed over time, so two characters with the same personality will behave the exact same way without any variations unless explicitly included. In order to present a believable complex personality, characters should be seen to make decisions based on their past experience via adaptation, and based on their perceived context.

1.1.4 Motivation from Games Domain

The model we seek to build will ideally contribute to the body of work in computer games. However, much of the academic work in emotions, personality, agents and learning is difficult to implement. Learning techniques are often complex for a lay-person to understand, can result in unpredictable behaviour, require significant computational power and require handcrafting of individuals. This results in making many models unsuitable for use in games.

Many recent computer games include large numbers of computer-controlled characters, such as in Grand Theft Auto, The Sims, and Oblivion. In many cases, these
1.1 Motivation

characters have been created using a limited number of handcrafted personality “types” or archetypes (Ellinger 2008). Every character with a particular personality type is essentially the same and not distinguishable from others of the same type, that is the characters are not individuals and appear homogeneous (Russell 2008). For example, the player can meet someone in a store in one part of town, move to another part of town and essentially meet the same character again, even though they are supposed to be different people. The longer a player spends in the world the more likely these similarities will be noticed and will decrease the player’s enjoyment of the game due to excessive repetition and predictability of behaviour.

A different approach is used in some other games, such as *Half-Life 2* where the player is required to work with a single computer-controlled character for much of the duration of the game. In *Half-Life 2*, the character is explicitly scripted so that its behaviour will change throughout the game. Unless this scripting is implemented by the designer, the character will not be able to adapt its behaviour over time and can result in the appearance of one-dimensional characters. In *Black and White*, the creature character can adapt, however it is only via the explicit teachings or reinforcements of the player, and not independently or in conjunction with its personality. In-game learning is probably only suitable for characters that the player interacts with for prolonged periods of time. However, learning prior to shipping the game will allow unique characters to emerge and will improve the diversity of characters a player meets, even if the player only has a short interaction with the character.

To be believable, characters should behave differently depending on their context. Within games, the characters can only recognise different contexts if the game developer has explicitly included that capacity with hand-scripting. For example, many characters are developed using finite state machines. If the character is in a particular state (e.g. surrounded by enemies), it will perform one action; if it is in another state, it will perform a different action. However, these differences must be hard-coded by the designer, who must consider each and every state and all the action possibilities available to the character. In our ideal model, the character would be able to learn which actions are appropriate for it personally on its own without enforced player interaction or hard-coding preferences.
1. INTRODUCTION

1.2 Research Questions

Personality can give life to virtual characters. However, current applications of personality require a large amount of handcrafting. We seek to reduce the amount of handcrafting required for a designer to build many individual characters. That is, this thesis aims to answer the question: How can characters with personality be created without handcrafting all behaviour? In addressing this question, we believe that cognitive-social personality theories offer a method of developing characters based on their individual experiences. Cognitive-social theories maintain that personality is both adaptive and context-aware (situated). By common definition, each personality is unique. So a requirement of personality itself is that characters should be individuals. These requirements lead to the following research questions:

1. How can a model of personality be created that uses adaptation? How does adaptation affect character behaviour?

2. How can a model of personality be created that uses context? How does context affect character behaviour?

3. How can personality be implemented so that the same template can be used to create a number of distinct, individual characters who behave differently?

These research questions can be broken down into a set of sub-questions relating to creating the model and testing the model: model-based and testing-based.

1. Adaptation:
   - Model-based sub-questions:
     - What aspects of personality can adapt?
     - How are decisions made?
     - How can characters calculate reward?
     - How can characters use reward to update behaviour?
   - Testing-based sub-questions:
     1.(a) Does behaviour change over time?
     1.(b) Can characters learn about specific, functional goals?
     1.(c) How does reward change with time?
     1.(d) What happens if adaptation is turned off?

2. Context:
   - Model-based sub-questions:
1.2 Research Questions

– How can context be represented?
– How can we fill in context information?

• Testing-based sub-questions:
  2.(a) Is character behaviour different in different contexts?
  2.(b) What happens if context is turned off?

3. Individual:

• Model-based sub-questions:
  – What is an individual within our model?
  – What is a personality template?
  – How does personality change over time (i.e. how can a character be different from another character with the same template)?

• Testing-based sub-questions:
  3.(a) Are the behaviours of characters different from each other over time?
  3.(b) Are any individuals obtained?

The model-based sub-questions can be answered by defining terms appropriately and implementing the model in an application. It is not possible to measure the validity of these subjective answers directly, the model itself is the “answer”. To determine whether the model developed is successful, we implement a game to be run with many different starting conditions, we consider sub-questions relating to testing and establish a set of criteria for success. From the runs of the game, we can extract data relating to three quantifiable measures of effectiveness: behaviour, reward and individuality. Behaviour relates to what the player can see on the screen, the actions that the characters choose to execute. In particular, we count the number of times that characters choose different actions over regular output time periods. Reward relates to how well the character is achieving their own goals according to their own personal evaluation. High reward values mean that the character is achieving its personal goals in the current environment. Individuality measures the number of differences between all characters in the game based on their behaviour. The maximum number for individuality implies that each and every character is completely different from every other character in the game, in the respect that they choose different actions at the same time periods.

Using behaviour, reward and individuality as measures of effectiveness of the model, led to a set of quantitative criteria for success the model should satisfy to be determined successful. Each criterion for success addresses a specific testing-based research sub-question as shown in Table[1.1]. We recognise that the choice of the majority as a cut-off
## 1. INTRODUCTION

<table>
<thead>
<tr>
<th>Research Questions and Testing-based Sub-questions</th>
<th>Criteria for Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How does adaptation affect character behaviour?</td>
<td></td>
</tr>
<tr>
<td>1.(a) Does behaviour change over time?</td>
<td><em>Behaviour</em> changes over time.</td>
</tr>
<tr>
<td>1.(b) Can characters learn about specific, functional goals?</td>
<td>When given a functional goal to learn, the majority of characters choose the “correct” action the majority of the time, based on <em>behaviour</em>.</td>
</tr>
<tr>
<td>1.(c) How does reward change with time?</td>
<td><em>Reward</em> values are on average higher using our model than when random choice is used.</td>
</tr>
<tr>
<td>1.(d) What happens if adaptation is turned off?</td>
<td>Compared to when adaptation is turned off, both <em>individuality</em> and <em>reward</em> are higher when adaptation is used.</td>
</tr>
<tr>
<td>2. How does context affect character behaviour?</td>
<td></td>
</tr>
<tr>
<td>2.(a) Does character behaviour differ in different contexts?</td>
<td>For one character’s <em>behaviour</em>, show that in different contexts the action chosen the majority of times is different.</td>
</tr>
<tr>
<td>2.(b) What happens if context is turned off?</td>
<td>Compared to when context is turned off; both <em>individuality</em> and <em>reward</em> are higher when context is used.</td>
</tr>
<tr>
<td>3. How can personality be implemented so that the same template can be used to create a number of distinct, individual characters, according to their behaviour?</td>
<td></td>
</tr>
<tr>
<td>3.(a) Are the behaviours of characters different from each other over time?</td>
<td>Character <em>behaviour</em> passes the chi-squared test.</td>
</tr>
<tr>
<td>3.(b) Are any individuals obtained?</td>
<td>Based on their <em>individuality</em>, at least one character is different from the majority of the other characters when they are all based on the same template.</td>
</tr>
</tbody>
</table>

*Table 1.1:* Criteria for success to be used to evaluate testing-based research sub-questions. Words in *italics* are the measures of effectiveness.
for some of the criteria is somewhat arbitrary. Having at least the majority of characters demonstrate a desirable behaviour shows that they are behaving as intended. We simply use the majority as initial criteria so that we can quantitatively rate the model that is designed and implemented in this thesis. Note that a chi-squared test measures whether the characters' behaviour (actions over time) are actually independent from each other.

1.3 Thesis Overview

This thesis begins with a literature survey of related work and introduces pertinent theories. This allows us to place our work within the broader research field and explain the theories that are used in our personality model. After the literature survey, we introduce our model for agent personality development, Chapter 3. This chapter details the key components that characters need, as well as the adaptation loop that characters use to make decisions and evaluate their choices. It shows how we use cognitive-social theories and somatic markers to develop personality that affects decision-making and evaluation. The model is generic and can be applied to many domains. We detail how to build individual characters and how to attach domain-dependent specifics to the generic model.

To test our model, we developed an example game domain. The game developed is based on simplistic school children who can insult each other and move around their world. Limited implementations have been shown to generate large numbers of possible paths or actions for characters ([Theune et al., 2004](#)). Within our simple game implementation, we were able to generate a large amount of complexity due to interactions between characters. In order to determine success of the model in terms of being able to generate different individuals from the same personality template, we use three measures of effectiveness: behaviour, reward and individuality. In Chapter 4 we introduce our implemented game, discuss our experimental setup, including how the measures of effectiveness are obtained, present our method for answering the research questions, the scenarios used and consider expected results.

In Chapter 5, we present our results to the testing-based research questions as well as some interesting side results that we found while answering the research questions. Results are based on the criteria for success and the measures of effectiveness. The side
results relate to domain-dependent aspects of our implementation. In this chapter we also discuss the implications of the results obtained.

In the final chapter results are summarised, particularly in relation to the research questions. We discuss future directions for research, implications for the game industry, and contributions made by our agent personality development model.
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 &
2. LITERATURE SURVEY

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 Theories

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
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]):

\[
C \left( \frac{SD-1}{S-1} \right)
\]

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 \( 3 \left( \frac{2^2-1}{2-1} \right) = 27 \). 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
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
2.1 Theories

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 emo-
tional 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 ap-
praisal process. Appraisal depends on goals, standards and attitudes (Ortony et al.,
1988). Variables that influence which emotion is triggered include desirability, praise-
worthiness and appealingness (Ortony et al., 1988). The intensity of the emotion gen-
erated depends on both local and global variables, such as reality, proximity, unexpect-
edness, 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 Theories

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. LITERATURE SURVEY

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
2.1 Theories

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

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).
- 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).
2.1 Theories

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.
2. LITERATURE SURVEY

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.
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 \cite{Damasio1994}. 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 \cite{Damasio1994}. 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 \cite{Damasio1994}. 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 \cite{Damasio1994}.

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 \cite{Lazarus1991}. Further, learning can give the appearance of personality \cite{Sanchez2004}.

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. LITERATURE SURVEY

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. $\epsilon$-greedy policies exploit the successful state-action pairs only some of the time, based on the parameter $\epsilon$. 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;
2. LITERATURE SURVEY

• 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]:

\[
p_{t+1}(a_t) = p_t(a_t) + \beta (r_t - \bar{r}_t)
\]

where \( r_t \) is the reward received on the last play and \( \beta \) is a positive step-wise parameter.

To update the reference reward, the following equation is used [Sutton & Barto, 1998]:

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

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).
2. LITERATURE SURVEY

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 $\epsilon$-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 \cite{Spronck2006} and hierarchical reinforcement learning \cite{Ponsen2006a}. They have also compared learning techniques for a simple problem within the RTS world \cite{Ponsen2006a}, and investigated ways to improve set up \cite{Ponsen2007} and speed \cite{Bakkes2006} 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 \cite{Spronck2006}. The characters should be able to be used against both beginners and experts \cite{Spronck2006}. Another aim is to reduce the complexity of the game and therefore allow the characters to learn more effectively \cite{Ponsen2007}, and for each character to optimise its learning selection policy \cite{Ponsen2006a}. Characters should be “interesting” \cite{Spronck2006}. 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 \cite{Ponsen2007, Spronck2006}. Ponsen et al. \cite{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 \cite{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. \cite{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 \cite{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 \cite{Bakkes2006}. Determining a suitable reward function when agents pursue multiple goals is difficult, as found in Ponsen et al. \cite{2006a}.

The agents learn domain-specific knowledge or rules about what can be done in the world \cite{Spronck2006}. They label state-action pairs with reward values \cite{Bakkes2006}. That is, observations (i.e. states) and action pairs are stored with an associated assessment of success/reward \cite{Bakkes2006}. After a new
2. LITERATURE SURVEY

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
2. LITERATURE SURVEY

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 Applications

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).
2. LITERATURE SURVEY

- Personality is based on a vector of six possible actively-pursued desires \cite{Parunak2006}.
- Personality includes OCC (cognitive appraisal) based goals, standards and preferences \cite{JohnsSilverman2001}, 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 \cite{DiasPaiva2005}.
- Personality is modelled using emotional monitoring, personality evaluation and behavioural transformation (i.e. capable of changing coping preferences based on past experience) \cite{Francis2010}.
- 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) \cite{Lim2005}.

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 \cite{DiasPaiva2005}. According to Lazarus, personality influences both appraisal and coping \cite{Lazarus1991}, 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, \cite{RousseauHayes-Roth1998} 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 \cite{Andre2009}, 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. \cite{Yingying2009}
### 2.2 Applications

<table>
<thead>
<tr>
<th>Aspect Influenced</th>
<th>Implemented/Proposed By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Appraisal</td>
<td>Gratch &amp; Marsella (2004); Silverman &amp; Bharathy (2005).</td>
</tr>
<tr>
<td>Decision-making (Secondary Appraisal)</td>
<td>Johns &amp; Silverman (2001) and Silverman &amp; Bharathy (2005); André et al. (1999).</td>
</tr>
<tr>
<td>Reward Calculation</td>
<td>Johns &amp; Silverman (2001); Yingying et al. (2002).</td>
</tr>
<tr>
<td>Goals or Desires</td>
<td>Parunak et al. (2005); Lim et al. (2005).</td>
</tr>
</tbody>
</table>

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.
2. LITERATURE SURVEY

Similarly in Gadanhó (2003), when making a decision, an initial emotional conclusion is made, which may then be rejected by a cognitive conclusion (Gadanhó, 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" (Gadanhó, 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).

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

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
2. LITERATURE SURVEY

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 \cite{Ahn & Picard, 2006}. The execution loop for each agent at each time step is \cite{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, \cite{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 \cite{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 \cite{Sanchez et al., 2004}. The system is deterministic, so actions always have the same reward consequences \cite{Sanchez et al., 2004}, which makes learning easier for the agents. Convergence of behaviour is not guaranteed \cite{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 \cite{Sanchez et al., 2004}.

Anticipation and Chromosomes \cite{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 \cite{Bozinovski, 2003}. Input personality is two traits (curiosity and patience) and a set of handcrafted “chromosomes” \cite{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 Velázquez (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 (Velázquez, 1998). The robot has simple plans to choose from and more than one action can be performed at the same time (Velázquez, 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
2. LITERATURE SURVEY

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).

### 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
2. LITERATURE SURVEY

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
2.2 Applications

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 char-
acters 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.
precisely 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 be-
haviour 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.
Chapter 3

Agent Personality Development Model

Our model is designed to enable complex personalities for character to be constructed without handcrafting every behaviour. In this chapter we explain how the model will satisfy all three research questions (see Section 1.2), page 14, by generating character personalities that are adaptive, context-aware and individual. The glossary on page xxiv provides a reference to the concepts and definitions used in this chapter.

Our model is designed to be generic and therefore applicable to any domain. However, to provide examples of the generic concepts, we use the motivating example introduced in Chapter 1. The motivating example is set in a world where characters live in villages. We will concentrate on one particular village and consider the characters within this village, since the model can easily be extrapolated to other villages within a more complex virtual world.

We begin by giving an overview of the model as if it was applied to the village example. Then we define the key aspects of behaviour and what personality means in our model. After this introduction we separate the components used to build the agents and the process that uses these components. This chapter addresses the model-based research sub-questions.

Illustration of Model as Applied to the Village Example The agents representing villagers have a number of activities they can perform including make, give, sell, buy, ask, use, interact and respond. The villagers are constantly doing one of
these activities and will automatically start a new activity after they have completed all other activities they were doing. The villagers have a number of soft goals that they can pursue (see literature survey Section 2.1.1.2, page 24), such as having friends, having money, not being hungry, and no one in the village being hungry. The villagers have no initial explicit knowledge of how to achieve these soft goals. Although many villagers can be pursuing the same goals, the importance or weight that any one villager places on each goal may differ. For example, one villager may place a high importance on not being hungry and a low importance on having money. The villager’s actions will affect achievement of their soft goals and they will need to learn which actions lead to achievement of the soft goals they personally consider important.

Making a Decision in the Village Example  When a villager is faced with the choice of which activity to do next, it starts by determining its current context, based on the current level of achievement of its soft goals. For example, the villager may be hungry and not have much money. They use this to look up the preference they have, based on past experience, for each of the available activities. For example, they use the context [not hungry and have money] to determine the preference for “give” and “sell” and any other available activities. Then the villager separates these activities into buckets of ‘desirable’, ‘undesirable’ and ‘don’t care’ according to the somatic marker hypothesis. In order to ensure all activities are explored appropriately, a random number is used to determine which group to choose an activity from first. This random number is compared to a cutoff value, so that it is highly likely that the ‘desirable’ group will be chosen but the other groups will also be occasionally chosen. If there are multiple activities within the chosen group, then, for simplicity, the villager randomly chooses an activity within the group to execute. This activity may require further decisions to be made. For example, if the chosen activity is to sell something then decisions about what to sell and who to sell it to need to be made. If this happens, then the decision process just described is repeated for those sub-activities or plans.

Evaluation in the Village Example  After the villager has finished executing its top-level activity (e.g. “sell”), it performs an evaluation of the activity and all the plans that were successfully executed. It determines how close it is to achieving each of its goals now which gives it an individual reward value for each goal. The villager
sums these values together, based on how important it considers each goal, to obtain an overall personal reward value for the activity. For example, the villager may still be hungry, but has more money. If this villager places more importance on not being hungry, the reward for this activity may not be as large as it could be compared to if making money was most important. The personal reward is compared to previous reward values to determine whether this activity was “good” (i.e. closer proximity to achieving all of its important goals) or “bad”. The reward is used to update the character’s preferences for the plans from which it chose in the context it was in when it made the decision. That is, if the villager choose to “sell food” in the original context, [hungry and have money], the preference for “sell food” may be decreased for this context, since the villager has not improved its achievement of goals. So the character updates its preferences and next time may choose a different choice since it has updated the preference on “sell food” for the context [hungry and have money].

**Building Relationships in the Village Example** Within the domain-dependent activities undertaken by the characters, they are able to build up relationships with other characters, for example, who they consider to be a “friend”. Over time the characters will acquire different sets of relationships with the other characters. Based on the decision-making and evaluation processes the characters will also have different preferences on activities depending on their experience. This will mean that each character is different from the others (individual in terms of observable behaviour), can adapt to new environments, and chooses activities based on its perceived context and past experience.

**Key Aspects of Behaviour** Personality of virtual characters can only be perceived by players or users based on actual behaviour exhibited by the characters within the world, as chosen by the agent part of the character. Based on cognitive-social theories (Bandura 1977) (see literature survey Section 2.1.2.2 page 32), and work by Ortony (2002) and Lazarus (1991) on individual differences (see literature survey Section 2.1.2.3 page 35), we believe that the causes of behaviour can be classified into three major categories:

1. A value and goal hierarchy, or a personality template;
2. Current context; and
3. AGENT PERSONALITY DEVELOPMENT MODEL

3. Past experience.

The personality template is used to guide learning and development to ensure that the personality is within appropriate limits. Context ensures that the agent’s behaviour is relevant to its current perceived situation. Past experience allows the agent to develop and learn. Note that our model works similarly to cognitive-social theories [Bandura, 1977] (see literature survey Section 2.1.2.2 page 32) where behaviour is influenced by the person and the environment; and the opposite is also true. The behaviour that an agent chooses will affect its environment and will affect its experiences, which will in turn affect future behaviour.

The personality template is constructed using a number of components that represent how the agent responds to the environment, the actions available and a ranking of importance of goals it is trying to achieve. It is generally accepted that both nature and nurture play key roles in personality development [Lazarus, 1991] (see literature survey Section 2.1.2, page 31). A personality template can be seen as an agent’s intrinsic nature, whereas its past experience can be seen as its nurture. The personality template can be considered to be similar to the value and goal hierarchy [Lazarus, 1991] considered to be a key component of personality (see literature survey Section 2.1.2.3 page 35).

In order for the agent to use context, the agent itself needs to store domain-dependent knowledge, such as objects the agent has, and who it likes. This domain-dependent information is required when calculating how close the agent is to achieving its goals. The achievement levels of goals are then used to determine what context the agent believes it is in.

We use Damasio’s somatic marker hypothesis [Damasio, 1994] (see literature survey Section 2.1.3 page 36) as the reinforcement learning selection policy to link an agent’s current context with its own past experience. We update somatic markers using an adaptation loop (reinforcement learning) that evaluates the success of activities (based on the agent’s goals), reward function, and creates corresponding somatic markers or preferences. The adaptation loop is the key element in the automatic development of agent personality.
3.1 Model Components

**Definition of Personality**  So, within the context of our model, what does “personality” mean and how does it influence behaviour? We take personality to mean the combination of the personality template and somatic markers (past experience). Somatic markers are used for decision-making, so that the choice of what to do now depends on the success of what was done last time this context was perceived. Thus, personality influences decision-making. Personality should also influence an agent’s evaluation of the success or failure of an activity. To model this aspect of personality, we use soft goal personality to evaluate success or failure of completed actions. Soft goal personality is the relative importance that an individual places on each of the available soft goals.

We will now discuss the agents’ components in our model and then we will detail how the agents use and modify these components within the adaptation process.

### 3.1 Model Components

In this section we investigate the major components that each agent needs to have in order to exhibit an individual personality that adapts and is context-aware. We address the following model-based research sub-questions:

- What aspects of personality can adapt?
- How can context be represented?
- What is a personality template?

The components or beliefs that each agent stores in our model are shown in Figure 3.1, we will discuss these in detail in the subsequent sections. We begin by an explanation of what soft goals are in our model and the main components they influence: achievement level, and context. Next, we discuss the somatic marker table and how somatic markers are stored and used. The three main components that make up the agent’s personality template are presented: goal/plan hierarchy, soft goal personality, and emotionality. We finish with a brief look at domain-dependent knowledge.

#### 3.1.1 Soft Goal Related Components

The goals that the agents are trying to achieve forms a key part of its decision-making and evaluation sub-processes. As explained in the literature survey, formal models of
3. AGENT PERSONALITY DEVELOPMENT MODEL

Figure 3.1: Agent components and beliefs

goals contain a number of different types of goals (Braubach et al., 2004) (see literature survey Section 2.1.1.2, page 24). Hard goals are directly achievable by implementation of a specific plan, according to the explicit goal/plan hierarchy. The set of adopted hard goals is non-conflicting and only one goal is pursued at a time. Soft goals work at a level above the goal/plan hierarchy and are a set of potentially conflicting goals that the agent is attempting to achieve at every step. The agent is not given any knowledge of which plans achieve their soft goals. Plans may contribute partially to achieving a number of different soft goals.

The agent is given a number of soft goals based on its soft goal personality, as part of the its fixed personality template. The soft goal personality specifies which soft goals the agent is pursuing, the importance it places on each of these goals, and the ideal goal value. The soft goal personality is described later in more detail as part of the personality template in Section 3.1.3.2 (page 80). The agents adapts and learns to find the best way (or ways) to achieve its soft goals.

We begin by examining the achievement levels of soft goals and the domain-dependent equations needed to calculate achievement levels. Then we define what context means in our model.

3.1.1.1 Achievement Level of Soft Goals and Soft Goal Equations

For each of the soft goals it is trying to achieve, the agent stores the current achievement level of that goal. For example, if the soft goal is “have friends”, then the achievement level represents a quantitative measurement of this. If the agent is friends with half of
the population of the village currently, the value may be set at 50%. To determine this
achievement level, domain-dependent soft goal equations are used. These equations
are functions that take knowledge of the world and determine the achievement level of
each soft goal. Achievement levels are updated during evaluation of an activity. The
agents use their current level of achievement of their soft goals to influence reward and
context calculations.

Storage of achievement levels is domain independent. However, since the actual soft
goals are domain-dependent, an equation to determine the achievement level must be
supplied. The soft goal equations are the same for every agent. That is, having friends is
measured in the same way by all agents, although the goal value may be different. Soft
goals must relate to the domain-dependent knowledge stored by the agent, otherwise
the equation cannot be used to measure success or failure of that particular soft goal.

3.1.1.2 Context

In this section we address the second research question (see Section 1.2, page 14):
How can a model of personality be created that uses context? We also address sub-
question: How can context be represented? We are interested in context because it
is a key component of personality according to cognitive-social theories and individual
differences (see summary in the literature survey Section 2.1.2, page 31).

We use context to relate to an agent’s soft goals, in particular, to describe overall
achievement level of all of its soft goals. For each soft goal, a letter is used to represent
the subjective achievement level. For example, if the agent is friends with half of the
population currently, its soft goal personality within its personality template will dictate
whether the soft goal “have friends” is considered to be ‘H’ (high), ‘M’ (medium), ‘L’
(low). If the achievement level is unable to be measured, it is assigned as unavailable
using ‘-‘. This can occur (for example) if the soft goal is “talk to friends” and the agent
has no friends currently. These letters are combined in alphabetical order of the soft
goal they represent. For example, if there are two soft goals “have friends” and “have
money”, a context of “HL” represents that the agent is currently in: have lots of have
friends and no money. The somewhat arbitrary distinction of ‘H’, ‘M’, ‘L’ is intended
to limit the total number of contexts possible.

It is possible to include further information within the context, if the goal/plan
hierarchy or domain requires it. This information is most likely to relate to decisions the
3. AGENT PERSONALITY DEVELOPMENT MODEL

agent has already made within the same top-level activity it is implementing currently. For example, if Chris has already decided to give away bread; when choosing “who to give it to” he should consider that he has already chosen to give bread. That is, the context could be “HL&Bread”. This extra information may not be necessary for many domains, particularly since it drastically increases the number of contexts about which the agent needs to learn.

By linking context to the achievement of soft goals, it provides a means for agents to learn that, when the achievement level of a particular soft goal is low, certain actions will improve the achievement of that goal. For example in the village domain, if Chris does not have many friends, giving away bread will probably help him gain more friends.

3.1.2 Somatic Markers

Damasio’s (1994) somatic markers (see literature survey Section 2.1.3 page 36) are a key concept in our model and are used as a simple method of allowing the effects of experience or gut instinct to contribute to decisions. Somatic markers are used to represent an agent’s past experience, so as to guide the agent’s decision-making towards or away from available plans, and can adapt with time. According to the hypothesis, somatic markers are linked to a physical body. However, we use computational methods to simulate the effect of somatic markers on decision making, not their physical effects.

In considering the research sub-question relating to which aspects of personality can adapt, somatic markers are a part of an agent’s personality and represent the aspect that adapts. During the adaptation process (see Section 3.2 page 82), the somatic marker preferences are used during decision-making and then updated based on the agent’s immediate experience.

In our model, a somatic marker is a record consisting of a context and plan key with a numerical preference value. When the agent is required to decide between more than one action in a context of which it has no experience, the agent creates initial somatic markers for every action that is applicable. After execution of an activity, each plan name and context pair that actually completed execution have the somatic marker preference updated for that particular agent, based on how successful the activity was deemed to be. The new preference is recorded and stored in that agent’s look-up table of somatic markers. This stored value is used the next time the agent is in that context and must choose between that particular plan and the others available. Calculation of
3.1 Model Components

the new, updated preference value is described in evaluation (see Section 3.2.2, page 90).

To explain how somatic markers work within the context of our model, let us take a concrete example from the village domain. Consider the agent Chris, who is trying to achieve the soft goals “have friends” and “have money”, and has equal importance on both. Chris, gets to a point where he can choose between the top-level activities. Chris uses his current achievement of soft goals to determine his context, for this example we will use “HL”, which means Chris has many friends and not much money. Then, Chris uses the context and each plan name to find all preference values for the plans he is choosing between. Based on these values he chooses a plan to try next (using the appraisal of coping process described in Section 3.2.1, page 87). Let us assume Chris has chosen “give”, he then implements that plan, which requires him to choose what to give and who to give it to, so he must look up the somatic markers for the available sub-plans. When Chris has finished the “give” activity he then performs an evaluation of the success of this activity and its chosen sub-plans in achieving his soft goals. Based on this the somatic marker preference values for the chosen plans and activity are updated for this particular context, i.e. “HL”.

If Chris has no friends, but a lot of money (“LH”), he should eventually discover that giving away products for free will improve his overall achievement of his goals, since he will gain friends. However, if Chris is friends with everyone, but has no money (“HL”), he should find that, giving away products for free will not improve his overall achievement of his goals. Chris will need to attempt other activities and may discover that selling his products will improve his overall achievement, only to the level that it does not take away too many friends. That is, Chris may learn that either giving or selling are his favourite activities depending on his context. In this sense somatic markers are similar to expected utility for plans within this context. However, if another agent, e.g. Adam, has the same importance on the soft goals “have friends” and “have money”, he may discover that to have friends and money he can interact with people and say nice things to them, buy objects from others, sell his products, and never give anything for free.

If there are many agents within the village who make bread (i.e. are bakers), then Chris may find it difficult to sell bread, since everyone will have bread of their own. In this circumstance, Chris may discover that if he makes candles, i.e. becomes a
3. AGENT PERSONALITY DEVELOPMENT MODEL

Figure 3.2: Possible partial goal/plan hierarchy from the village domain. Part A.

candlestick maker, he will be able to sell his candles and make more money. This is a preliminary step to generating observable personality. Being a baker or a candlestick maker are not usually considered to be personality traits, so this concept needs to be followed further. When the agents are given more diverse choices, such as singing while making bread, or yelling while selling products, then the agents could be seen to choose, not just what to do, but how to do it in their own way. Agents learn these preferences automatically based on past experience with the other agents and the environment itself.

3.1.3 Personality Template

In research question three we ask how personality can be implemented so that the same template can be used to create a number of distinct, individual characters, according to their behaviour. This in turn leads to the question: What is a personality template?

Our goal is that the developer can create a small number of templates, and yet end up with a much larger number of observably different agents who still have some similar goals and behaviour. For the agent component of each character, we define a fixed personality template (potentially the same as that of a different agent) that helps guide evaluation, decision-making and agent behaviour so that the agent does not adapt and change uncontrollably. The personality template cannot change over time, even though the agent’s observed personality, in the form of behaviour, can change over time due to learning. The personality template restricts the way that the agent’s overall “personality”, including somatic markers, can develop. The personality template can be handcrafted to suit many agents. There are three components to the
3.1 Model Components

Figure 3.3: Possible partial goal/plan hierarchy from the village domain. Part B.

personality template: goal/plan hierarchy; soft goal personality; and emotionality. We now explain each of these in turn.

3.1.3.1 Goal/Plan Hierarchy

Our model uses the BDI agent programming paradigm (see literature survey Section 2.1.1.1, page 21). In a BDI paradigm, the plans available to an agent are represented in a goal/plan hierarchy. The goal/plan hierarchy is designed for the particular domain and represents the hard goals (not soft goals) and the associated plans an agent can use to achieve the goals and indirectly its soft goals. Goal/plan hierarchies have a hard goal at the top and the plans indicate ways to achieve this goal. The hierarchy is designed so that each goal in the hierarchy has at least one plan that will directly achieve that goal. In simple cases, by successfully executing a plan that handles the goal, it is assumed that the hard goal has been achieved, without testing to confirm this. Plans are able to post sub-goals that are all required to be achieved for the plan to be successful. At the bottom level are plans with simple steps that do not require any sub-goals. One possible (partial) implementation of a goal/plan hierarchy for the village domain is illustrated in Figures 3.2 and 3.3. In this example implementation, eight top-level activities are available for an agent to choose from: Make, Sell, Give, Ask, Buy, Use, Interact, and Respond.

For a given domain, it is likely that a large number of different agents can use the
same goal/plan hierarchy. Many domains will only require one goal/plan hierarchy as long as it is rich enough to encapsulate all actions that agents can execute. In our model, the domain-dependent goal/plan hierarchy is linked into a generic goal/plan hierarchy to ensure that the agents can perform an evaluation of their choices. This generic structure is described in the adaptation process (see Section 3.2 page 82).

3.1.3.2 Soft Goal Personality

The agent’s soft goal personality represents the soft goals that the agent seeks to achieve, the importance of each goal, and the level it must reach to be considered “achieved”. This enables the agent to calculate its own personal reward values (i.e. self-reinforcement) during the evaluation process, rather than relying on an external trainer (as most reinforcement learning problems require, see literature survey Section 2.1.4.2 page 38). Our agents are able to determine personal reward based on what they consider to be “good” and “bad” according to the soft goals they are trying to achieve.

In hierarchical reinforcement learning (RL) (see literature survey Section 2.2.1.3 page 49), every agent has the same goals. These goals are put into a hierarchy by the designer so that the agents can determine reward when they have multiple goals. In our model, the agents are given a list of the soft goals they should achieve as part of their input personality template.

As described earlier, soft goals can be conflicting. For example, the agent may simultaneously want to have objects and give away objects. What can be different for each agent is the importance, or weight, it places on each soft goal; whether it wants to maximise or minimise the value; and the ideal value of the goal. For example, Chris may place a very high importance on making friends, and a small importance on making money; while Adam agent may place a high importance on no one being hungry. The soft goal personality includes information on what the agent considers to be the maximum or ideal value for each soft goal. For example, one agent may consider “have money” achieved when it has $100,000, another agent may only consider the soft goal achieved with $1 million. If any of the soft goal personality values are not provided for a particular soft goal, the domain-dependent default is used.
3.1 Model Components

3.1.3.3 Emotionality

The term emotionality is used to represent an agent’s emotional personality, similar to Ortony [2002] (see literature survey Section 2.1.2.3, page 35) and includes aspects that Velásquez [1998] used for temperament (see literature survey Section 2.2.2.3, page 59). Different individuals will react to the same event differently based on their emotionality.

In Figure 3.1 (page 74), it can be seen that emotionality is made up of a number of different values. We define emotionality as a vector of values:

1. uptake values \( (u) \): how the initial reaction to an emotion event is recorded. A high negative uptake value means the agent considers even the smallest bad event to be very bad. A low positive uptake value means positive events will not affect emotions as much;
2. decay values \( (b) \): how long it takes for a particular emotion event to decay with time. Low decay values mean that the agent will take a long time to forget an event;
3. domain-dependent thresholds: cutoff values for emotions, opinions etc. Used by the agent to determine if something or someone is “good” or “bad”;
4. soft goal achievement thresholds: cutoff values for allocating a letter (e.g. ‘H’, ‘M’, or ‘L’) to a soft goal achievement level. This will determine the agent’s context.
5. bucketing thresholds \( (δ) \): used in decision-making to determine which plans are “desirable” (see Section 3.2.1, page 87). High bucketing thresholds mean that a plan must be very good to separated out from other plans and be considered “desirable”;
6. bucketing choice threshold \( (τ) \): used in decision-making to determine which bucket to choose from (see Section 3.2.1, page 87). A low bucket choice threshold will cause the agent to be more likely to exploit successful plans, rather than explore other choices; and
7. step-wise learning parameters \( (α, β, r_0) \): used to moderate the influence of recent rewards compared to past rewards in learning;
3. AGENT PERSONALITY DEVELOPMENT MODEL

3.1.4 Domain-dependent Knowledge

The agents must store information about their domain in order to be able to evaluate the achievement of their soft goals and therefore evaluate their success or failure. In the village example domain, beliefs (knowledge) that the agents store could include: amount of money they have, objects they currently hold (e.g. have 2 candles), happiness and other emotions, attraction values towards others and how much others are attracted to them.

3.2 Adaptation Process

The previous section introduced the main components of each agent. In this section we address how these components are used and modified so that the agent can adapt. This relates to the first research question as to how can a model of personality be created that uses adaptation. In doing so, we consider the following model-based sub-questions:

- How are decisions made?
- How can characters calculate reward?
- How can characters use reward to change behaviour?
- How can context information be provided?
- How does personality change over time (i.e. how can a character be different from another character with the same template)?

In our model, personality includes past experience, as implemented by somatic markers. An agent’s somatic markers are used to assist decision-making and adapt over time based on self-calculated rewards. Characters calculate reward based on their soft goal personality and experiences in the environment. The personality template guides how they adapt, but each agent will have its own experiences, that generate its own unique somatic markers. In this way, personality is enabled to change over time and allow agents to differ from each other, even given the same initial personality template.

The agents use an execution loop at every time step and an adaptation loop for every activity (i.e. longer duration). We begin this section by explaining the difference between the execution loop and the adaptation loop in our model. Then we detail the exact steps of appraisal of coping choices, followed by evaluation.
3.2 Adaptation Process

**Figure 3.4:** Expanded version of the BDI execution loop: The standard BDI execution loop expanded to use appraisal of coping choices based on somatic markers.

**Figure 3.5:** Adaptation process: triggered by an incoming or self-generated event.
3. AGENT PERSONALITY DEVELOPMENT MODEL

Execution Loop  At every time step the agents follow an execution loop, see Figure 3.4. The model we are proposing adds a more complex decision-making process to the plan choice step of the standard BDI execution cycle (compare Figure 3.4 to the standard cycle shown in Figure 2.2, page 24). The extra step is appraisal of coping choices which uses the somatic marker hypothesis (Damasio, 1994) (see literature survey Section 2.1.3, page 36), the steps of appraisal of coping choices are explained in Section 3.2.1 (page 87). Appraisal of coping choices corresponds to secondary appraisal in the cognitive appraisal model of emotions (see literature survey Section 2.1.1.3, page 27) and acts as the reinforcement learning selection policy, the function that maps from perceived states of the environment to action (Sutton & Barto, 1998) (see literature survey Section 2.1.4.2, page 40). Note that if there are no new events, then the agent simply executes the next step in its current plan, if it has one.

In order to develop personality automatically, a learning feedback (adaptation loop) is needed to update each agent’s past experience, represented by somatic markers. That is, in order to learn how to improve the appraisal of choices process, a reward function needs be used in conjunction with a value function. This adaptation loop requires that the agent uses a specific adaptation process, see Figure 3.5. In this process, the agent chooses a plan to execute, executes it, and then evaluates the success of this plan with respect to achieving its soft goals. The process becomes a loop because after every evaluation the agent will automatically choose something new to do, if it is not currently doing anything else.

At every time step the agent completes the execution loop shown in Figure 3.4. A single step in a plan would probably not have changed the agent’s perception of the environment sufficiently for the agent to be able to determine the effect and merit of that single step. For example, it would be difficult to perform an evaluation after a small step, such as choosing who to talk to. This means that the adaptation process (Figure 3.5) needs to apply only to major actions (called activities) and constitutes a higher level loop than the BDI-based execution cycle. For example, a suitable activity would be a longer interaction, such as actually having an entire conversation with an agent. A generic structure is needed to force the agents to follow this adaptation loop in addition to the standard execution cycle.
3.2 Adaptation Process

Figure 3.6: The adaptation process with a generic goal/plan hierarchy: linking the adaptation loop to the top-level of a domain-dependent BDI hierarchy.

Adaptation Loop The adaptation process (Figure 3.5) is enforced by combining some domain independent goals and plans and raising them to the top of the goal/plan hierarchy, see Figure 3.6. This ensures that an agent will always evaluate its choices. There are two built-in plans that the agent can use: a proactive plan and a reactive plan. The reactive plan is triggered by an incoming event (goal) from the environment (e.g. new agent arrived in village), from another agent (e.g. a request to interact or to buy something), or from the agent itself (e.g. choose what to bake). The proactive plan is used when the agent has no activities to do currently, and this also functions to ensure agents are always doing “something”. Both plans cause the agent to perform the following adaptation process shown in Figure 3.5 and described here.

1. Appraisal of coping choices: Choose between activities or plans to decide what to do next.
2. Execution of the chosen plan: this might involve posting another goal and then making further appraisal choices of sub-plans. Repeat steps 1 and 2 as required by the domain-dependent goal/plan hierarchy.
3. Evaluation: after the top-level activity has been finished, evaluate achievement levels of soft goals; evaluate personal reward; update somatic marker values (preferences); and update emotions.
3. AGENT PERSONALITY DEVELOPMENT MODEL

Figure 3.7: Effect of adaptation processes on components and beliefs: How components and beliefs affect and are affected by the two main adaptation process, i.e. appraisal of choices and evaluation.

This adaptation process is similar in some respects to the execution loop used by Ahn & Picard (2006) to increase the efficiency of learning and decision-making by using emotions in the decision-making process (see literature survey Section 2.2.2.2, page 58). In summary, we have enhanced the standard BDI architectures by including somatic markers, soft goals, and evaluation. This is done by adding an extra step when choosing which plan to implement and forcing agents to evaluate results after top-level activities have been completed.

Components and the Adaptation Process Figure 3.7 shows the two main subprocesses of the adaptation process, appraisal of choices (decision-making) and evaluation, and how they affect the agent’s components. Personality can be observed in actions done, which can be seen to be influenced by somatic markers in appraisal of choices, and these in turn are influenced by soft goal personality in the evaluation process. In appraisal of choices, the domain-dependent goal/plan hierarchy and context are used to look up the somatic markers for available actions. This leads to a choice between the actions. In evaluation, the actions taken, domain-dependent evaluation functions (soft goal equations) and the ideal soft goal value from the soft goal personality are used to calculate a value of how much each soft goal is being achieved currently (achievement level of soft goals). This in turn leads to an update of the agent’s context. The domain-dependent soft goal equations also lead to an overall personal reward value when combined with the importance of soft goals from the soft goal personality. The
reward value is used to update the somatic marker preferences during the evaluation of a completed activity.

We will discuss in detail Steps 1 and 3: appraisal of choices and evaluation. Step 2, execution of a plan, is domain-dependent based on the designer defined goal/plan hierarchy.

3.2.1 Appraisal of Coping Choices

In this section we address more fully the research sub-question relating to how decisions are made. The appraisal of coping choices process is used for decision-making between multiple plans. Coping in the cognitive appraisal model of emotions (see literature survey Section 2.1.1.3, page 27) is the result of a reaction to a new emotional state generated by a new event and the primary appraisal process. In our model we use coping as plans or actions that the agent can use to improve their overall wellbeing in the form of achieving their soft goals (not just their emotional state).

The appraisal of coping choices process is part of the execution loop performed at every step and also part of the adaptation loop. In the execution loop, appraisal of choices is used to determine which plan to execute next. Appraisal of choices fulfills the role of the reinforcement selection policy and is considered part of adaptation because the agent needs to remember what it choose so that it can update the policy based on its experience. In the cognitive appraisal model, appraisal usually refers to the initial (primary) evaluation made when an event is recognised and emotions are updated. Secondary appraisal in the cognitive appraisal model is the decision-making step and is used to decide what to do next to cope with the emotions generated. We use the term appraisal of coping choices to refer to this secondary appraisal process. The third part of appraisal (according to the cognitive appraisal model) is reappraisal which is implemented in our model as evaluation (see Section 3.2.2, page 90).

The appraisal of coping choices process is based directly on Damasio’s somatic marker hypothesis (see literature survey Section 2.1.3, page 30). The hypothesis describes how humans make decisions in the real-world. Although we do not simulate physical bodies for our agents, the appraisal of coping choices process is otherwise consistent with the somatic marker hypothesis.

When a goal is posted that could be handled by more than one plan, the agent uses appraisal of coping choices to determine which plan to choose to execute. If the chosen
3. AGENT PERSONALITY DEVELOPMENT MODEL

plan fails to finish successfully, the BDI system automatically reposts the original goal. This causes the agent to repeat the appraisal of choices process and choose from the other applicable plans, excluding the failed plan. When only one plan is left, there is no need to follow the appraisal of choices process since no choice can be made. The final plan is implemented, but will not have its somatic marker updated since the plan was not chosen. If the final plan fails and there are no more applicable plans available, then the goal will fail. This may cause the plan that posted that goal to fail and so on up the goal/plan hierarchy, unless this failure is handled explicitly. The steps in the appraisal of coping choices process in our model are described as follows:

1. **Determine which plans are applicable.** Examine the goal/plan hierarchy to see which plans are available to handle the goal posted, then consider whether each plan is currently applicable. For example, the plan “give something” is not applicable if the agent has nothing to give. This step generates a list of applicable plans.

2. **Find the current context using soft goal achievement levels.** Use the “H”, “M”, “L” value for the current level of achievement for each of the soft goals and place together in alphabetical order of the name of each soft goal (see Section 3.1.1.2 (page 75). If the domain requires, add any extra information to expand the context. For example, if the agent is choosing who to give an object to, the type of the object itself may need to be included.

3. **Obtain somatic marker preference.** For each applicable plan, $a_i$, find the current somatic marker preference, $p_{s,t}(a_i)$, for this context, $s$, at the current time, $t$, using the somatic marker lookup table. If this context-plan pair has not been encountered before, set the preference to an initial domain-dependent value.

4. **Bucket plans.** Group plans into three buckets based on their preference values, $p_{s,t}(a_i)$, to simulate the somatic marker hypothesis (Damasio, 1994) (see literature survey Section 2.1.3, page 36) that will guide decisions towards or away from plans. To group plans, we use the average preference, $\bar{p}$, and standard deviation of preferences, $\sigma$, for applicable plans. We also use the agent’s bucketing thresholds, $\delta_{-ve}$ and $\delta_{ve}$, from emotionality in the personality template. The plans are grouped into “desirable”, “undesirable”, and “don’t care” based on:
3.2 Adaptation Process

- if \( p_{s,t}(a_i) > \bar{p} + \sigma \delta_{+ve} \); plan is in the “desirable” group;
- if \( p_{s,t}(a_i) < \bar{p} - \sigma \delta_{-ve} \); plan is in the “undesirable” group; and
- if \( \bar{p} - \sigma \delta_{-ve} \leq p_{s,t}(a_i) \leq \bar{p} + \sigma \delta_{+ve} \); plan is in the “don’t care” group.

5. **Decide which group to use.** If the agent always chooses a plan to execute from the “desirable” group, then the first plan that reaches the desirable group will always be chosen and the agent will cease to explore other possibilities. To handle the trade-off between exploration and exploitation, a random number, \( \epsilon \), is chosen, where \( 0 \leq \epsilon < 1 \). This value determines which group the agent should attempt to choose from first. The bucket choice threshold value from emotionality, \( \tau \), guides the use of \( \epsilon \) according to:

- if \( \epsilon \geq \tau \), try to choose in the order:
- if \( \epsilon \leq \frac{1}{4} \tau \), (i.e. a small number and very unlikely to occur), try to choose in the order:
- else, try to choose in the order:

In this way, the “desirable” group is chosen most often, but not always. The other groups are chosen less frequently, but the potential for exploration is retained.

6. **Choose a plan randomly from the first group chosen.** If there are no plans available in the preferred group, attempt to choose a plan from the next group and repeat until a plan is chosen. Based on the somatic marker hypothesis [Damasio, 1994] (see Section 2.1.3, page 36), at this point a more thorough and in-depth domain-dependent decision-making process can be used to determine which plan should be chosen from within a “desirable” group. This is because somatic markers are meant as a coarse sorting of all plans available so that the in-depth analysis does not need to be performed on a potentially large number of plans. However, we wanted to test whether somatic markers improved decision-making at all and decided to begin with a simpler model, using random choice within the group to reduce domain-dependent coding.
3. AGENT PERSONALITY DEVELOPMENT MODEL

Usually in RL, when preferences for plans are obtained the selection policy is simply a probability function to determine how likely it is for each plan to be chosen. However, we are using the somatic marker hypothesis which only allows us to determine whether a plan is “good” or “bad”. This means that our process takes the stored (continuous) somatic marker value and turns it into a discrete value by bucketing it.

3.2.2 Evaluation

In this section we focus on the research sub-questions: How can context information be provided? How can characters calculate reward? How can characters use reward to change behaviour?

When making decisions, an agent uses its current context in the appraisal of coping choices process to determine which plan to choose for this particular configuration of soft goal achievement levels. Depending on its soft goals, an agent will have a number of different contexts, although it is only in one at a time (see Section 3.1.1.2, page 75). For each context, the agent needs to know its somatic marker preference, i.e. what to do in this context. In previous applications that use context when making decisions, this detail has been handcrafted by the designer (e.g. Campos et al. 2006; Rousseau & Hayes-Roth, 1998) (see literature survey Section 2.2.2.1, page 55). The main goal of this thesis is to create individual characters with a minimum of handcrafted behaviour. So, we use evaluation as a self-reinforcement learning process to automatically build up the somatic markers that an agent needs for every context it encounters. In this evaluation process, the individual (not an external trainer) calculates for themselves how much they prefer one outcome over another (Cloninger 2008; Phares & Chaplin 1997) (see literature survey Section 2.1.2.2, page 32).

To give feedback on the activity completed and therefore enable adaptation, the agent performs an evaluation of how well it is achieving its soft goals. The results (i.e. level of success or failure of plans) are stored so that they can be used by the agent to encourage the agent not to make the same mistake when it is next in that context deciding what to do. As found by Ponsen et al. (2006a), determining a reward function when there are multiple goals can be difficult (see literature survey Section 2.2.1.3, page 47). We use a process that combines how important each individual agent considers its goals. This leads to the evaluation process consisting of the following steps.
3.2 Adaptation Process

1. Update achievement levels and calculate the individual reward for each soft goal.
2. Calculate personal reward, how well this agent thinks it is doing currently and the success of the plans executed.
3. Update somatic marker preferences for all the plans that were successful, using the context from when the choice was made.
4. Update emotions.
5. Update the reference reward.

Each of these steps is now described in more detail. That is, how personal reward (self-reinforcement value) is calculated and how the somatic marker value for a given context is updated.

3.2.2.1 Step 1: Individual Soft Goal Reward Calculation

For soft goal, \( j \), at time \( t \), the achievement level, \( z_{t,j} \), is updated by the domain-dependent soft goal equations. For example, if the soft goal is to “have money”, then the amount of money the agent believes it has now can be used as the measure. As another example, if the soft goal was “not to be hungry”, then the achievement level could be based on the amount of food the agent has eaten over the last time period.

Since the soft goals themselves and the beliefs (facts and opinions) that the agent stores are domain-dependent, the calculation of the achievement level is necessarily domain-dependent.

Most beliefs are updated directly within the domain-dependent plans of the goal/plan hierarchy. For example, if a plan gives the agent money, the agent’s fact about how much money it has will be updated as a step within that plan. Opinion-related beliefs are also updated in this way, e.g. if the plan caused the agent to change its opinion of another agent. Some soft goals are based on the agent’s emotions, such as “be happy”. Unlike other beliefs of the agent, emotions are updated based on the agent’s personal reward value. This occurs as one of the final steps during evaluation, see Step 4. This means that, if the soft goal depends on an emotion, the individual reward value calculated here (in Step 1) would reflect what the agent did in the previous cycle, not the current one, since emotion has not been updated yet. To overcome this, the agent uses an estimate of what the current emotion will become based on the achievement level of all other soft goals and the previous emotion value.
3. AGENT PERSONALITY DEVELOPMENT MODEL

A domain-dependent reward function, based on the distance to the ideal soft goal value, $Z_{j,\text{max}}$, is used to calculate reward for each soft goal. The ideal value can be set within the soft goal personality or the default domain-dependent value can be used. The absolute reward, $r_{\text{ABS}t,j}$, at time $t$ (i.e. now) for each soft goal, $j$, is:

$$r_{\text{ABS}t,j} = Z_{j,\text{max}} - z_{t,j}$$  \hspace{1cm} (3.1)

where $z_{t,j}$ is the current (just calculated) achievement level of soft goal, $j$; and $Z_{j,\text{max}}$ is the desired goal value of the reward. The range of this absolute reward is $0 \leq z_{t,j} \leq Z_{j,\text{max}} - Z_{j,\text{min}}$, where 0 is the best value the absolute reward can take.

This method assumes that the maximum value is greater than the current value. However, if, for example, the soft goal is for no one in the village to be hungry, the ideal value would be that no one is hungry, i.e. 0. In this circumstance, the absolute reward should be calculated using

$$r_{\text{ABS}t,j} = z_{t,j} - Z_{j,\text{max}}$$  \hspace{1cm} (3.2)

This gives an absolute reward on the same scale and with 0 being the best value.

Reward is commonly used where high values represent a desirable reward, negative values are punishment, and zero is neither a reward or punishment. So to convert reward so that the range is $r_{\text{min}} \leq r_{t,j} \leq r_{\text{max}}$ (in many domains it is likely the range will be $[-1, 1]$), the following equation is used when ideal value is greater than the current value (after substituting in equation $3.1$):

$$r_{t,j} = r_{\text{max}} - (r_{\text{max}} - r_{\text{min}}) \times \frac{Z_{j,\text{max}} - z_{t,j}}{Z_{j,\text{max}} - Z_{j,\text{min}}}$$  \hspace{1cm} (3.3)

The above equations are suitable when the agent wants to maximise the soft goal, e.g., “have friends”. However, sometimes the agent wants to minimise the goal, e.g., minimise “have enemies”, which is different from placing a low importance (weight) on the goal “have enemies”, which would mean that the agent does not care whether or not it has enemies. For this case, the equation to calculate reward would be converted using Equation 3.2 instead of Equation 3.1. All agents use the same soft goal equations to work out the individual soft goal reward values, but the beliefs they hold are likely to be different. Therefore the reward value for a specific soft goal is almost invariably different.
3.2.2.2 Step 2: Calculate Personal Reward

This step answers the research sub-question as to how can characters calculate reward. That is, how do the agents calculate their own personal reward value (self-reinforcement)? After all the rewards for each individual soft goal have been determined in Step 1, the agent’s overall personal reward for the activity under evaluation needs to be calculated. The personal reward reflects how close the agent is to achieving all of its soft goals. Some goals may be more or less important to an agent based on its soft goal personality (set in the personality template) and so the rewards are weighted to reflect this. The value is called personal reward because each agent will evaluate the events from their perspective and generate their own reward, i.e. a personal reward.

To explain the reasoning behind this process, let us consider the example of Adam and Chris who are both pursuing the same soft goals, “have money” and “have friends”, and have the same ideal value for the goals. During the last activity both these agents received a large amount of money, but they still had no friends. This means that the agents both have the exact same achievement levels for the soft goals, and therefore the same context and individual reward values. If Adam and Chris had the exact same soft goal personality, then their personal reward values would be exactly the same (assuming the same result from the last activity, which is itself unlikely). If their soft goal personalities, i.e. weights on soft goals, are different, then their personal rewards would likewise be different. Adam’s soft goal personality may place a high importance (weight) on “have money” and a smaller importance on “have friends”. On the other hand, Chris has the opposite soft goal personality: low importance on “have money” and high importance on “have friends”. In this case, based on the activity just completed where they gained money and no friends, Adam would calculate his personal reward as being very high, whereas Chris would have a lower personal reward. This reflects the fact that Chris considers activities to be “good” when he gains friends. Note that Chris is still trying to gain money and this gives him some reward, it is just that he prefers to have friends.

Accordingly, the personal reward, \( r_t \), is found using a weighted sum of the soft goals derived from the weights of the soft goal personality:

\[
 r_t = \frac{\sum r_{t,j} \cdot w_j}{\sum w_j}
\]  

(3.4)
3. AGENT PERSONALITY DEVELOPMENT MODEL

where \( w_j \) is the weight the agent places on soft goal \( j \). The weight is a value from \([0, 1]\), where 1 represents the highest importance for this goal, and 0 means no importance whatsoever. The scale for personal reward is \( r_{\min} \leq r_{t,j} \leq r_{\max} \). In many domains it is likely the range will be \([-1, 1]\).

3.2.2.3 Step 3: Update Somatic Marker Preferences

This step addresses the research sub-question relating to how characters use reward to change behaviour. This is the learning step of the adaptation process. To update somatic markers, the agent must decide whether the personal reward received constitutes “success”. We use a simple reinforcement learning technique, the reinforcement comparison technique, from Sutton & Barto (1998) (see literature survey Section 2.1.4.2 page 40). The technique compares the current reward, \( r_t \), with a reference reward, \( \bar{r}_t \), in order to update the preference for choosing a particular plan. The reference reward is based on all previous rewards and is used to determine whether the current reward obtained is “good” in comparison to all past rewards received in all contexts and for all activities. For example, a reward of 0.8 might be “good” if previous rewards were 0.5. However, if previous rewards were all 0.95, then 0.8 may be considered “bad”.

The somatic marker preference, \( p_{s,t}(a_i) \), for a plan, \( a_i \), selected during the last play (at time \( t \), when the agent was in context \( s \)) is incremented so that at time \( t + 1 \) (i.e. the next time step) the preference, according to the theory Sutton & Barto (1998) (Equation 2.2 page 40), will be calculated as:

\[
p_{s,t+1}(a_i) = p_{s,t}(a_i) + \beta(r_t - \bar{r}_t)
\]

where \( \beta \) is a positive step-wise parameter and \( \bar{r}_t \) is the reference reward. The reference reward is an incremental average of all recently received rewards for all activities in all contexts. After updating the somatic marker preference, the reference reward is updated and according to Sutton & Barto (1998) (Equation 2.3 page 40) this is calculated as:

\[
\bar{r}_{t+1} = \bar{r}_t + \alpha(r_t - \bar{r}_t)
\]

where \( \alpha (0 < \alpha \leq 1) \) is a step-size parameter, as used in many other reinforcement techniques.

In our model, plans may be ranked at varying levels within the goal/plan hierarchy. Plans that are very low in the hierarchy have only a small number of different ways
of executing that plan. For instance, in our example goal/plan hierarchy in Figure 3.2 (page 78), the plan “choose to make bread” has no sub-plans and can only be executed one way. However, the plan “make something” has a number of ways that it can be executed: by making bread, meat or candles. Each of these plans forms one possible plan path for “make something”. Evaluation is performed after an activity has been completed, i.e. when a single path (potentially consisting of many plans) has been executed. However, the activity could have been completed by following a different path. For each plan that was executed, the preference should be adjusted based on the possible paths below this plan in the hierarchy. This reflects that the other paths that were not taken may have a different reward and as a result this plan may not be “good” or “bad”, it could have been this particular chosen path was “bad”. So, “make something” may not be a bad plan just because “make bread” did not achieve the soft goals. The agent may need to attempt “make candles” instead. In order to include this, the preference is updated by a factor of \(1/b\), where \(b\) is the number of possible paths below this plan.

In some domains, it is possible for a plan to be repeated multiple times before evaluation is performed. For example, if the agent is interacting with another, it might post the goal “choose something to say” multiple times before the activity, i.e. conversation, is over. If this happens, the particular plan considered may only have been chosen a small number of times from the total number of times that there was the opportunity to choose this plan. To include this weighting, the preference is updated using the ratio of number of times chosen to number of times it could have been chosen.

By combining these factors, the somatic marker preference for choosing plan \(a\) at the next time step \(t+1\) in context \(s\) is updated according to:

\[
p_{s,t+1}(a_i) = p_{s,t}(a_i) + \beta \cdot d \cdot (r_t - \bar{r}_t)
\]

where

- \(d\) is a factor calculated per plan based on the number of paths below this plan, \(b\), and the number of times this plan could have been chosen in the last activity, \(c\), out of the total number of times the plan could have been chosen, \(c_{TOT}\). This accounts for the plan’s position in the goal/plan hierarchy and the fact that a
3. AGENT PERSONALITY DEVELOPMENT MODEL

plan may have been executed more than once within a single activity:

\[ d = \frac{c}{c_{\text{TOT}} \cdot b} \]  \hspace{1cm} (3.8)

- \( \beta \) is a step-wise parameter from emotionality. It is usually between 0.2 and 0.8;
- \( p_{s,0}(a_i) \), initial reward is set based on the domain; and
- \( \bar{r}_t \) is the reference reward.

Only plans that have been chosen and passed are updated. Note that the process gives preferences that are unbounded, unless the designer explicitly restricts them.

3.2.2.4 Step 4: Update Emotions

In the cognitive appraisal model (see literature survey Section 2.1.1.3, page 27), emotions are updated based on primary appraisal immediately after an event has been received. In systems with an emphasis on emotion modeling, this can be a very complex process, based on a domain-dependent analysis of individual beliefs. In our model, we update emotions only during evaluation (called reappraisal in the cognitive appraisal model) after an activity (coping) as been performed. In our implementation we use a single emotion, “happiness”, that is based on the overall personal reward calculated. This happiness reflects a simple “good”/“bad” analysis of how the agent feels. If other emotions are required for the domain, they can be used similarly to happiness, although the emotion event update may be triggered at a different time, rather than after evaluation.

Every time an evaluation is performed, a happiness emotion event, \( k \), is generated. An emotion event consists of an initial intensity, \( I(k, t_0) \), the time at which the event happened, \( t_0 \), and a decay value, \( b \). The initial intensity is determined by:

\[ I(k, t_0) = r_t \times u \]  \hspace{1cm} (3.9)

where \( u \) is the emotion uptake value from emotionality in the personality template. If the reward was positive, the positive uptake value is used, and if negative, the negative uptake is used. This reflects that some people may react more strongly to positive events compared to negative events. For example, an agent might be very upset if they lose something, but only mildly happy if they gain a lot.
3.3 Summary of the Model

Emotions decay over time. After a given time, $t$, the intensity, $I$, for a given event, $k$, can, according to Dias & Paiva (2005) (see literature survey Equation 2.4, page 54), be calculated as:

$$I(k, t) = I(k, t_0) \times e^{-b(t-t_0)}$$  \hspace{1cm} (3.10)

where $b$ is the decay value from emotionality in the personality template.

The overall current emotion value is determined by summing the intensities of all relevant emotion events at the given time of evaluation:

$$I(t) = \sum_k I(k, t)$$  \hspace{1cm} (3.11)

If $I(k, t)$ falls below a small threshold, then the event is discarded, i.e. forgotten, for future calculations.

3.2.2.5 Step 5: Update the Reference Reward

The reference reward, $\bar{r}_t$, is updated according to the reinforcement comparison technique (Sutton & Barto, 1998) (see literature survey Equation 2.3, page eqnLitSurvey:RefRewardUpdate) using:

$$\bar{r}_{t+1} = \bar{r}_t + \alpha (r_t - \bar{r}_t)$$  \hspace{1cm} (3.12)

where

- $\alpha$ is a step-wise parameter from emotionality;
- $\bar{r}_0$ is the initial reference reward and is set to a value based on the domain.

The agent stores only one reference reward to be used in Step 3 to determine whether the personal reward value is “good” or “bad”. The reference reward is independent of the particular activity that is being evaluated currently and context used.

3.3 Summary of the Model

The model presented in this chapter represents a reduced version of the cognitive-social view of personality. Characters use self-reinforcement to adapt to their environment, that is, change themselves and their behaviour. An agent’s behaviour within the environment can change the environment itself and this may, in turn, change the agent,
similar to the process shown in Figure 2.3 (page 33). According to cognitive-social theories, expectancies allow people to predict what the world will be like in the future (Cervone & Pervin, 2008) (see literature survey Section 2.1.2.2, page 32). Somatic markers take on a similar role. When deciding between different activities, an agent uses its somatic markers to predict how choosing a specific activity will affect its soft goals.

In our model, agents store past learning experience as somatic markers. These somatic markers guide the agent’s future decisions towards or away from certain choices. Agents use a soft goal personality to specify the goals they are trying to achieve and to determine how to calculate their personal reward (self-reinforcement value). An agent’s soft goal personality is part of its personality template. This personality template also includes emotionality and the hard goals and plans it can use to achieve its soft goals (i.e. goal/plan hierarchy). The personality template guides the development of an agent’s somatic markers, and therefore guides the development of the agent’s personality.

### 3.3.1 Generic Infrastructure Implementation

This model can be implemented entirely generically, so that it can be re-used in many application domains. Implementation was performed using the JACK development environment (Howden et al., 2001) that provides automatic support for BDI architectures (see literature survey section 21, page 21). The core goals and plans required for enforcement of the adaptation loop were set up so that any domain-dependent goal/plan hierarchies could link into them, as shown in Figure 3.6 (page 85). A simple top-level program can be run to link together the generic and domain-dependent aspects. Each domain-dependent plan needs to have additional steps included automatically so that the appraisal of coping choices process can be followed where necessary and information stored about which plans are chosen and in which context. The domain-dependent equations, to determine achievement level and reward for individual domain-dependent soft goals, are stored in a single area. The structure to determine current context, update somatic markers and emotions can be built without any knowledge of the actual domain to be used.
3.3.2 Building an Individual

In this section, we answer the research sub-question: what is an individual within our model? An individual is a character who is different from other characters in their reasoning, i.e. in the agent part of the character, not its physical appearance. This difference is manifested in observed behaviour (executed actions) of the agents. In our model, the agents choose actions based on their somatic markers. So the major factor in making an individual is their somatic markers. There are a number of other components that make up an individual in our model, see Figure 3.1 (page 74). We summarise the components as follows.

- Personality:
  - Somatic markers.
  - Personality template:
    - Goal/plan hierarchy,
    - Soft goal personality,
    - Emotionality.
- Current context.
- Achievement level of soft goals.
- Domain-dependent knowledge (beliefs), including facts and opinions.

To build an individual in our model, there are a number of steps to follow. Firstly, the generic model must be added to by deciding on domain-dependent knowledge and soft goals to be provided for all the agents. Then, the designer creates a small number of personality templates, each of which can be used by more than one agent. Setting up a goal/plan hierarchy can take a substantial amount of time so, to ease the burden on handcrafting, we suggest that many agents can have the same goal/plan hierarchy. Some agents can have the same soft goal personality and emotionality. The individual agent emerges from this over time by using the adaptation loop to incorporate its unique experience. The components that become unique to a particular agent are its somatic markers, current soft goal achievement levels, current context, and personal knowledge (beliefs). This distinction is illustrated in Figure 3.8.

The number of possible starting personality templates depends entirely on the domain and time available for the project. If many very diverse characters are required, more goal/plan hierarchies could be developed. To obtain smaller character differences
Figure 3.8: What makes an individual an individual in our model: Components that only one agent has compared to components and processes that all agents have.

(but still significant), more soft goals should be possible to use in the soft goal personality. The range of emotionality parameters is also domain dependent, but the number of parameters is fixed to the list shown in Section 3.1.3.3, page 81.

According to Kluckhohn & Murray (1953), personality theories attempt to understand and describe why each person is in certain respects: like all other people, like some other people, like no other people (see literature survey Section 2.1.2, page 31). In our model, all agents use the same adaptation loop, some agents have the same personality template; and after time no agents have the same somatic markers, and therefore no agents will have the same behaviour.

This chapter has addressed the model-based research sub-questions throughout. The exact listing of the questions with precise answers is provided in the concluding chapter. The model we have presented will now be tested for its success at delivering individual, adaptive and context-aware characters. In order to test the model, an implemented version of the model and a testing methodology was required to be built prior to obtaining results.
Chapter 4

Implementation and Experimental Setup

Our generic model presented in the previous chapter was explained with assistance from examples of the village domain. However, the model can be applied to any domain. In this chapter, we introduce and explain the particular domain created to test our model. We used a prototyping method of testing where we developed a pilot implementation followed by the final implementation. The use of a pilot implementation was intended as a proof of concept and was used to improve the model design itself and guide the construction of the final implementation. The final implementation was used to test our research questions and generate quantitatively measurable results to address the criteria for success.

In this chapter, we begin by describing the pilot implementation and its findings. We follow this by describing the final implementation domain covering: the goal/plan hierarchy used for all characters, context labels used, the domain-dependent knowledge and, finally, soft goals and how achievement levels of soft goals were determined. After this, we detail experimental set ups and considerations. We begin with initial findings relating to updating somatic markers and then introduce the emotionality variables used as determined by preliminary testing. To test the criteria for success relating to the research sub-questions we used three main measures of effectiveness: behaviour, reward and individuality. We outline how we obtained these quantitative measures, then

---

1Examples from the actual implementation were not used in the previous chapter so as to assist separation of generic and domain-dependent aspects.
we introduce the scenarios used to test starting conditions on personality templates. Finally we discuss expected results for each of the criteria for success for the scenarios concerned.

4. IMPLEMENTATION AND EXPERIMENTAL SETUP

Choosing an implementation domain to test the model was very difficult. The style of games that would benefit most from the model are (complex) social games where (human) players interact for extended periods of time with large numbers of virtual characters. We settled on a simple domain relating to school children making friends. Although simplistic, it was thought that, since the characters could generate their own friendships, they would be able to exhibit relatively complex behaviour. The more complex the domain, the harder it would be to determine what the characters were doing and develop appropriate quantitative measures to evaluate their performance.

By showing that the model works to some degree in a simple domain, this would indicate it would be even more effective in a more complex domain. This is because eventually we seek to allow human participants to see the complex structure called “personality”. If something as complex as personality is visible in a simple domain where the characters do not have many choices between what to do, then, in a more complex domain, there would be more differences to observe. That is, a complex domain would allow the characters to choose between more plans and therefore differ from each other in more ways, and personality would potentially be more visible.

Often in games and virtual agents, testing is based on user feedback and questionnaires. When testing a new model, it is advisable to make sure the model is reaching its criteria for success before using human participants. In this way, we can consider a larger number of variations of time periods, emotionality and personality templates than would be feasible to ask a human participant to test and provide feedback.

We considered the best way to test the model without human participants would be in two stages: a preliminary pilot model to test the concepts, and a second stage to test the criteria for success. By doing a pilot study, we were able to test the model itself, refine the pertinent concepts and determine which variables have the greatest effect on behaviour. The pilot model allowed us to improve the model before our final, more in-depth, implementation.
4.1 Implementation

In this section, we begin by outlining the pilot implementation and the findings obtained. After this, we introduce the game used for the final implementation. We present the goal/plan hierarchy and outline how context is used in this particular domain. We introduce the domain-dependent knowledge that characters need to store to enable them to function within this game. The domain-dependent knowledge leads us to list the soft goals available to characters. We finish by documenting how soft goal achievement levels were calculated in this particular domain.

4.1.1 Pilot Implementation

We begin by briefly outlining the pilot domain before discussing the most important lessons we learned from the pilot implementation. The pilot implementation was based on school children characters having conversations together (Sandercock et al., 2006). The characters could choose who to talk to and have “conversations” (passing topic knowledge sentences between each other). Characters were given some initial knowledge and preferences on topics, and were then expected to learn about topics from other characters and also whether they were interested in a topic. The pilot implementation relied less on adaptation than the model presented in Chapter 3 and required characters to have initial knowledge. This meant the characters still needed more handcrafting than desired because characters had to have a lot of initial beliefs or knowledge in order to be able to react to the world. To address this problem, the final model and implementation did not require prior knowledge for the characters.

Although characters were using somatic marker preferences for their top-level activity choices, somatic markers were not used further down the goal/plan hierarchy, such as when choosing what to say within a conversation. This meant that preferences on what the characters should say within a conversation were hard coded by the designer. This process involved creating a laborious preference table that was the same for every character. This problem was solved in the final implementation by allowing characters to learn somatic preferences for all plan choices, and not just the top-level activities.

All characters were given only one soft goal to achieve “be happy”. The point at which they considered this achieved was based on how well three factors were achieved: “like emotions”, “like conversation topics”, and “like having friends”. The characters placed weights on these each factors to represent how important they considered that factor. This weight was learnt by the characters themselves during an initialisation process.
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

phase and fixed after initialisation. As a result, characters who were talking about
topics they liked were more likely to want to keep doing this; resulting in much less
diversity of character behaviour than was desired. By varying both the character’s
appraisal of choices and their evaluation method, the characters could become unstable,
and were much more prone to have difficulty learning anything outside their previous
experience, i.e. they did not explore all possibilities enough.

The evaluation function was not suitably complex and, if the character was un-
happy, this led to a bad evaluation, which led to a more unhappy character and so on.
Evaluation and emotion were so entwined that once the character became slightly
unhappy, they would continue until they were very unhappy and unable to find any
path to get out of the cycle. To reduce this problem, it was considered best to fix
the parameters used in one of appraisal of choices (decision-making) and evaluation.
Evaluation was chosen to be fixed, so that somatic markers could develop and change.
To address the evaluation problems, the final model used a more complex evaluation
based on more than hapiness by incorporating soft goals to define their goal state.

Results from the pilot implementation indicated that the characters had some di-
versity, but measurement presented a problem. Characters were clustered slightly, but
the clusters were only loosely subjectively detectable and a quantitative measure was
not developed. There were problems with the interface between the character AI and
the GUI, which resulted in non-repeatable runs. Therefore, in the final implementation,
this was monitored very closely to ensure repeatability. The pilot implementation iden-
tified some gaps in the initial concepts in the model and enabled these to be rectified,
so as to arrive at the model presented in Chapter 3.

4.1.2 Game Description

After analysing the pilot implementation, the model was refined, enabling the next
stage of the testing: creation of a final implemented game to provide the basis for
answering the research questions. The final implemented game is also based on social
interactions in the form of building friendships between characters. The domain is
similar to the pilot, but instead of extended conversations, the characters pass only
insults and, in addition, characters can move or wait around. The characters are a
group of school children, such as those in Figure 4.1, on their lunch break who want to
interact with each other. Individual children can choose from three different top-level
4.1 Implementation

Figure 4.1: Three characters from the game: Anna, Bec and Chloe.

activities: to wait in one place, move towards or away from another character, and tell another child the name of a child they do not like, i.e. insult someone. The children choose actions in real-time.

A character’s personality template includes their goal/plan hierarchy, emotionality and soft goal personality (see Section 3.1.3, page 78). Since the aim of this thesis is to determine whether a small number of templates can lead to personality diversity, we made all characters use the exact same goal/plan hierarchy (shown in Figure 4.2). At the top of the hierarchy is the generic goal/plan hierarchy structure that is required for the model to be able to adapt appropriately (see Section 3.2, page 82 and Figure 3.6). Underneath the generic plans and goals, each character can choose from four possible activities. If the character is responding to an incoming insult, they can use the plan “listen”; otherwise, they have a choice of “move”, “wait around” or “insult”. After a character has listened to another character (“listen”), they reply by agreeing or disagreeing with the incoming insult. When a character is pro-actively choosing what to do, it has 22 different possible paths through the goal/plan hierarchy. That is, there are 6 different ways the character can “move” (e.g. move away from an enemy; or move towards a friend); 15 ways the character can “insult”; and one way to “wait around”. In Figure 4.2 external messages represent the sending of insults and replies between different characters. If a reply is not sent to an insult, then it is assumed that the listening character disagrees with the insult.

The implemented game can be seen in Figure 4.3. If a (human) player is in the game, they act as another child and have access to the same information and actions as the characters. In Figure 4.3, the player’s avatar has “bunny ears” and is being told an insult about another character. The player has the opportunity to agree or
Figure 4.2: Implemented domain-dependent goal/plan hierarchy used for all characters.
4.1 Implementation

Figure 4.3: Screenshot from the implemented game: Anna tells the player an insult about Fran. Colour of speech bubble matches to the box around the character’s name.

Figure 4.4: Screenshot from the implemented game: Anna insults the player. Colour of speech bubble matches to the box around the character’s name.

disagree with the statement. In the same Figure we can also see some other characters interacting via an insult.

Each child character is trying to achieve a number of domain-dependent soft goals, such as making friends, not being insulted, and being close to friends. When a character gets an interaction request from another character, such as Figure 4.4, they respond immediately and perform an evaluation on the choices they make, even if they are doing something else at the same time. These screenshots with the (human) player are to show how the game could be played. For quantitative testing, there is no player interacting with the characters, see Figure 4.5. Excluding players allowed multiple runs to be executed under different starting conditions without player variability affecting
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

Figure 4.5: Screenshot from the implemented game without a player included. The captions in grey were used for debugging purposes.

the quantitative measures.

Within the domain-dependent actions that the characters execute, they are able to build up relationships with other characters, i.e. who they consider to be a “friend”. After a long period of time, each character will have a different set of relationships with the other characters and will also have different preferences on actions for different internal contexts. This process will mean that each character is different from the others (individual), it can adapt to new environments and chooses actions based on its perceived situation. The characters’ fixed soft goal personality encourages coherent behaviour. For example, one character may find that the best way to make friends is to wait around until someone talks to them (another might make friends by telling one character they do not like another character) but both characters are still trying to make friends.

The game was constructed because it appeared simple enough to show some complexity, but not so complex that the different personalities would be difficult to distinguish from other confounding factors. The complexity is evidenced by the different friendships the characters build with each other and the way they can execute the different activities. When choosing an activity pro-actively, the characters can choose between 22 different possible paths. In addition to this, from the viewpoint of each character, there are seven other characters in the environment, some of which are friends and enemies. So in fact, the number of different paths a character can take is substan-
4.1 Implementation

Initially more than 22. For example, Chloe can choose to move towards her friend, Bec, and not towards another friend, like Elle.

The characters begin with no inherent preference for or against any of the actions. A character will choose which action to perform based on its somatic marker preference values, initialised to the same neutral starting value. The restriction to allow only insults (and not compliments) forces the characters to choose who they will be friends with more specifically, and means that characters are not able to be friends with everyone, enhancing the potential for the characters to become different. Further, the characters are not able to actively make friends with other characters, they need to learn the indirect means of making friends, for example, insulting a character that is not liked by anyone.

To react to the world, a character needs to store domain-dependent knowledge (beliefs) about their world, including their opinions towards the others in the world. These beliefs enable the character to calculate achievement levels for its soft goals of this domain. In the following sections, we begin by outlining how context was used in this game domain, introduce the domain-dependent knowledge that the characters were able to store and describe how the domain-dependent knowledge is updated. Then we detail the domain-dependent soft goals used for this game and the equations for calculating achievement levels for these soft goals.

4.1.2.1 Context

In most instances, the characters use simple contexts, being the achievement level string without any additional information. For example, if the character had two soft goals to achieve, one possible context is “HL”. Based on the goal/plan hierarchy, Figure 4.2 (page 106), there were three exceptions to this.

- Choosing what to say in reply to someone else, i.e. “choose insult response”. The context is the achievement levels plus whether the insult is towards itself, an enemy, neutral, or a friend, e.g. “HL&Friend”.
- Choosing who to move with respect to. The context here is the achievement levels plus the direction the character has chosen to move, i.e. towards or away, e.g. “HL&Away”.

109
Choosing what to say and choosing who to insult. The context here is the achievement levels plus who the character has chosen to tell the insult to, i.e. friend, enemy, neutral, e.g. “HL&Neutral”.

4.1.2.2 Domain-dependent Knowledge

The goal/plan hierarchy and soft goals were developed in conjunction with the domain-dependent knowledge the characters could store and the game domain itself. This domain-dependent knowledge is not considered as part of the model because it relates only to the particular game scenario in the final implementation. However, it is important to specify exactly what knowledge the characters could have and how they update their knowledge because this knowledge affects achievement levels of soft goals, which in turn affects the somatic markers and the emergence and development of personality.

We divide the domain-dependent knowledge (beliefs) into two categories: facts and opinions. Facts relate to beliefs that the character has been given by others or the environment itself and which the character cannot change the values of on its own. Opinions relate to beliefs that the character has created itself, values placed on objects or other characters. Opinions can be changed by the character themselves. An example opinion is that Anna can consider Elle a friend and then change her mind if she so chooses. In the final implemented game, the characters had five groups of beliefs they could store related to: insults, location, happiness, attraction from others and attraction towards others.

**Facts: Insults** When the character insults someone or is told an insult they store the information along with whether or not both parties agreed to the statement. This allows the character to remember what has happened, so they can determine achievement levels of soft goals, such as “don’t be insulted”. The beliefs are stored as: “A insulted B and told C; C agreed/disagreed”. Only characters who are either the speaker or the listener (i.e. one of A or C) can store information. Characters are not able to overhear other character’s insults or pass on insults from others.

**Facts: Location** The agents are embodied as characters within the domain using avatars and so must store information about where they are within the domain. They also store information on which characters are close or far away from them. That is,
for every other character in the domain, they store a discrete value of the location that can be “close” or “far”. This allows the characters to calculate achievement levels for location related soft goals, such as “be close to friends”.

**Facts: Attraction From Others** The beliefs associated with attraction are how much the character likes the other characters and how much the other characters like this particular character. Characters are told by the other characters when they become a “friend” or “enemy”. This means that characters can store discrete information on the attraction other characters have towards themselves. This information is used by the characters to update their attraction to others. For example, if Chloe knows that Elle likes her, then Chloe will be more likely to like Elle in return. When Chloe changes her attraction of Elle from “neutral” to “friend”, she will send a message to Elle to tell her of the change. This way every character can keep a coarse track of who does or does not like them. Note that although this is an attraction value, the character cannot change its value directly. That is, if Chloe knows that Elle does not like her, then this is a fact in Chloe’s eyes. Chloe can only change the value, so that Elle likes her, if Elle tells her that herself.

**Opinions: Attraction Towards Others** In addition to keeping information about how other characters feel towards them, an individual character stores information on how much they “like” or “dislike” every other character. Attraction of others is updated within individual plans in the goal/plan hierarchy (Figure 4.2, page 106). Particularly, within the plans “insult” and “listen to insult”. The plan “insult” is where a character chooses someone to tell an insult to, whereas “listen to insult” is where the character has been told an insult by someone else.

In Section 3.2.2.3 (page 94), we described the reinforcement comparison technique used to update somatic marker preference values. When considering how to update attraction values, we used rather complex methods, inspired by the reinforcement comparison technique, to try to elicit some of the complexity that is associated with how people may decide who is their friend in the real world. The way that attraction is updated depends on whether the character is telling the insult or receiving the insult. The new attraction value is based on the previous value plus or minus a factor that depends what has been said and about whom it has been said. Decision flow charts
can be used to determine how to update attraction, as illustrated in Figures 4.6 and 4.7. Figure 4.6 is used when the character has insulted someone; Figure 4.7 is used when the character has been told an insult from someone else. Generally, the characters like being spoken to and when another character agrees with what they have said. The characters are able to update their attraction to the other character in either of the two plans. When a character is listening to an insult from another character (Figure 4.7), they may also change their attraction to the character that is being insulted, depending on how strongly they feel about the characters involved.

Let us consider an example where Anna tells Chloe an insult about Bec. Both Anna and Chloe will update their opinions of each other and Chloe may also update her opinion of Bec. To determine how Anna updates her attraction of Chloe, we use the process shown in Figure 4.6. Anna needs to answer the following questions.

- Did Chloe accept my (Anna’s) statement?
- Is Bec my (Anna’s) enemy?
- Is Bec my (Anna’s) friend?

For example, if Chloe accepted Anna’s statement and Bec is Anna’s enemy, then Anna will increase her opinion of Chloe (the enemy of my enemy is my friend). Chloe uses the process shown in Figure 4.7 to determine how to update her opinions of both Anna and Bec. If Anna has insulted Chloe to her face or insulted one of Chloe’s friends, then Chloe would decrease how much she likes Anna. If Chloe had no strong opinion of Bec currently, then she would update her opinion of Bec based on how much she likes Anna. If Anna is her good friend, then she will believe the insult about Bec and vice-versa.

The equations used to determine exactly how much to increase or decrease the attraction values are shown in Figures 4.6 and 4.7. Each equation differs due to the complex interaction of the three characters involved in an insult: the speaker, the listener and the character being insulted. Some preliminary testing allowed us to determine the values of the constants in the equations. We used $\alpha = 0.2$, so that old opinions matter more than new opinions. When increasing or decreasing the attraction values by a fixed amount, the value $\delta = 0.25$ is used. This value is based on being a small proportion (1/8th) of the attraction range of $[-1, +1]$. When updating how much the character is attracted to another character, the opinion of the other character is

---

1The equations are not repeated in the text because it is difficult to understand the equations if they are not placed within the context of when to use them according to the flowchart.
4.1 Implementation

I just told C: I insult B

Did C accept my statement?
No

Is B my enemy?
No

Is B my friend?
No

Decrease attraction to C
$\omega_{M,C,t+1} = \omega_{M,C,t} - \alpha \left( -\frac{1}{2} \omega_{M,B} + \kappa \cdot \omega_{C,M,B} \right)
$

Decrease by large amount, since I do not like B.

Increase attraction to C
$\omega_{M,C,t+1} = \omega_{M,C,t} + \alpha \left( \delta + \kappa \cdot \omega_{C,M,B} \right)$

I like people who don’t like insulting my friends, but don’t like being rejected.

Do nothing.

Do not want to insult friends, but like being agreed with.

Decrease attraction to C
$\omega_{M,C,t+1} = \omega_{M,C,t} + \alpha \left( -\delta + \kappa \cdot \omega_{C,M,B} \right)$

Decrease by large amount, since I do not like B AND I like being agreed with.

Increase attraction to C
$\omega_{M,C,t+1} = \omega_{M,C,t} + \alpha \left( \delta + \kappa \cdot \omega_{C,M,B} \right)$

I like being agreed with, but do not really care about B.

Figure 4.6: Flowchart used to determine how to update attraction values when the character (“Me”) has insulted someone. Terms used:

$\omega_{A,B,t}$ is how much A is attracted to B at time $t$;

$\omega_{C,M,t}$ is based on the discrete value of how much the other character (C) likes this character (Me).

Since it is discrete, the value used here is based over a period of time (i.e. “I have been liked for a long time”);

$\alpha$ is the same step-wise learning parameter as used to update the reference reward;

$\delta$ is a small value; and

$\kappa$ is a constant weight on the importance the character gives to the other character’s attraction value.
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

Figure 4.7: Flowchart used to determine how to update attraction values when the character (“Me”) has been told an insult. Terms used:

- $\omega_{A,B,t}$ is how much A is attracted to B at time $t$;
- $\omega_{C,Me}$ is based on the discrete value of how much the other character (C) likes this character (Me).

Since it is discrete, the value used here is based over a period of time (i.e. “I have been liked for a long time”);

$\alpha$ is the same step-wise learning parameter as used to update the reference reward;

$\delta$ is a small value; and

$\kappa$ is a constant weight on the importance the character gives to the other character’s attraction value.
taken into consideration. For example, when Chloe is updating her attraction towards Elle, she includes how Elle feels towards her, i.e. Chloe. This “attraction from” value is stored as a discrete value, but is calculated over a period of time to determine how long the other character has been feeling that way (e.g. Elle has believed Chloe is a friend for a long time). The “attraction from” value is multiplied by a constant weight of $\kappa = 0.1$ to reduce its affect.

**Opinions: Happiness** In our domain, we had a single emotion, happiness. In our implementation, it is based on the personal reward values calculated and therefore relates to the character’s achievement of its soft goals. Each time a reward is calculated an emotion event is generated. This process is described in Section 3.2.2.4 (page 96).

### 4.1.2.3 Soft Goals

Although the storage of soft goals is domain independent (see Section 3.1.1, page 73), the actual soft goals, and how to calculate achievement levels, are domain-dependent. For example, in the village domain, one possible soft goal could be “have money”. This soft goal would not be applicable in our implemented domain since the characters are not able to obtain money by any means. In the following list, the soft goals for our final game implementation are described along with information on how the achievement level of the soft goal, $x$, can be calculated. The achievement levels are all based on ratios, so the maximum value the soft goal achievement level can be is +1, and the minimum is zero. The achievement levels are based on the domain-dependent knowledge introduced in the previous section. In the following list, we specify which knowledge is necessary to be able to calculate the associated soft goal achievement level. These achievement values are calculated during the first step of the evaluation process (see Section 3.2.2.1, page 91).

- **Be Happy**: Uses the belief *happiness*. Achievement is determined based on the current intensity, $I$ (see Section 3.2.2.4, page 96), of happiness compared to the maximum value that happiness could be.

- **Make Friends/enemies**: Uses the beliefs *attraction from others* and *attraction towards others*. To calculate the achievement of the goal “make friends”, the character uses, for every other character in the simulation, a combination of
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

<table>
<thead>
<tr>
<th>Character A</th>
<th>Like (+1)</th>
<th>Don’t Care (0)</th>
<th>Don’t Like (-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Like (+1)</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Don’t Care (0)</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>Don’t Like (-1)</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
</tbody>
</table>

Table 4.1: Calculation of friendship points for two characters: used to determine achievement level for the soft goals “make friends” and “make enemies”.

whether they like a particular character and whether that character likes them. When a character likes another character, that counts as +1 points. When a character does not like the other character, that counts as −1 points. Table 4.1 shows how to work out the friend points for each relationship between character A and character B. Equal weight is given to both characters’ opinions. The character calculates the points for every possible relationship it can have with every other character in the game and generates a total, \( n_{\text{total}} \). The maximum possible number of friend points the character can have is \( n_{\text{max}} = 2 \times n_{\text{characters}} \), where \( n_{\text{characters}} \) is the number of other characters in the simulation. This value represents the circumstance where this character likes everyone and everyone likes it. Using these values, the character can work out its current achievement level, \( z \), using a ratio of how many points they have currently to the maximum (ideal) value:

\[
z_{\text{friend}} = \frac{n_{\text{total}}}{n_{\text{max}}} \tag{4.1}
\]

The character uses a similar equation to determine achievement of “make enemies”, counting enemies rather than friends.

• Insult Everyone/Enemies/Friends: Uses insults beliefs. The achievement of this soft goal is based on the last \( N \) activities completed. This allows a character to forget old insults. The measure takes into account insults towards the group specified in the soft goal name, i.e. everyone, enemies or friends. There are three different types of insults.

1. Mutual insults (both parties agreed), \( n_{\text{mutual}} \).
2. Insults that this character said, but the other character did not agree with, \( n_{\text{Isaid}} \).
3. Insults that the character was told, but did not agree with, \( n_{\text{theySaid}} \).
4.1 Implementation

The total number of “insults” is found by adding the insult types using a weighted sum, so that mutual insults are considered to be a stronger insult than non-mutual insults. The following equation is used to calculate achievement level, $z$:

$$z = \frac{w_{\text{mutual}} \times n_{\text{mutual}} + w_{\text{Isaid}} \times n_{\text{Isaid}} + w_{\text{theySaid}} \times n_{\text{theySaid}}}{N}$$ (4.2)

The exact weights used in the simulation were: $w_{\text{mutual}} = 1$, $w_{\text{Isaid}} = 0.5$, and $w_{\text{theySaid}} = 0.2$.

- Don’t be insulted: Uses insults beliefs. Measured based on the number of times this character was personally insulted, $n_{\text{insulted}}$ over the last $N$ activities (regardless of whether it was agreed to or not):

$$z = \frac{n_{\text{insulted}}}{N}$$ (4.3)

- Be close to everyone/enemies/friends: Uses location belief. Measured using the number of enemies (or friends or everyone) that are close, $n_{\text{close}}$ and the total number of enemies/friends/everyone, $n_{\text{total}}$:

$$z = \frac{n_{\text{close}}}{n_{\text{total}}}$$ (4.4)

If the soft goal is based on friends/enemies and the character does not currently have any friends/enemies, then the achievement level cannot be calculated.

- Meet everyone: Uses attraction towards others belief. Measured using the number of characters this character has met, $n_{\text{met}}$ and the total number of other characters in the simulation, $n_{\text{total}}$:

$$z = \frac{n_{\text{met}}}{n_{\text{total}}}$$ (4.5)

This soft goal was used for testing to confirm that the characters could successfully achieve a simple, easily obtainable soft goal.

Soft goals relating to insults were based on the last $N$ activities. In our implementation, we used $N = 5$. This number was chosen after initial testing of the final implementation showed the characters were not recovering from insults that happened a long time in

\footnote{If an attraction value for a character has not been initialised, then the character has not been met yet.}
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

the past. In a future implementation, it may be useful to investigate whether a decay function could be used so that recent insults are considered more important than insults that happened a long time ago.

4.2 Experimental Setup and Considerations

We now explain precise implementation details used to obtain the answers to the research questions and the associated criteria for success, listed in Table 1.1 (page 16). We begin by outlining the simulation running conditions. Then we discuss a problem discovered during preliminary testing relating to how the somatic marker preferences are updated in our domain. We then specify the emotionality used for all characters as part of their personality template. To address the criteria for success, we identify three quantitative measures that can be used to test effectiveness of the theoretical model: behaviour, reward and individuality. We will discuss how data on each of these measures will be gathered in our implementation. To obtain results, we used five Cases based on varying the soft goal personalities of the characters. We present these Cases as well as discussing how to test the effect of adaptation and context on the quantitative measures of effectiveness. We finish this chapter by considering the expected results from these tests in relation to the criteria.

The experiments were implemented without a (human) player and were completely repeatable. Based on preliminary results, runtime was fixed at a time where the characters appeared to be mostly in the same contexts and somatic marker preferences were relatively stable. Behaviour and reward data is obtained based on output data of reward and somatic marker beliefs. Every time a character evaluated its personal reward, the value was output to a file to store the information, see Figure 4.8. At set output times, the somatic marker beliefs of each character was dumped in separate files, see Figure 4.9. The data is an agent’s somatic marker JACK-belief in JACOB format. In Figure 4.9 “preferenceValue” is the agent’s current somatic marker value and “updatedCount” is the number of times this particular context-plan pair has been updated so far in the simulation. The “updatedCount” represents the number of times

\[^1\text{See JACK documentation for details. Accessible through } \url{http://www.agent-software.com.au/products/jack/documentation_and_instructions/index.html}\]
4.2 Experimental Setup and Considerations

<table>
<thead>
<tr>
<th>Time, EvalNum, Reward, RefReward</th>
</tr>
</thead>
<tbody>
<tr>
<td>6057, 646, 0.3214285714285714, -0.2995583838991689</td>
</tr>
<tr>
<td>6066, 648, -0.34523809523809523, -0.20641034060000785</td>
</tr>
<tr>
<td>6067, 647, 0.3214285714285714, -0.22723450379572097</td>
</tr>
<tr>
<td>6083, 649, -0.34523809523809523, -0.14493504251207712</td>
</tr>
<tr>
<td>6093, 650, -0.34523809523809523, -0.17498050042097985</td>
</tr>
<tr>
<td>6103, 651, -0.34523809523809523, -0.20051913964354715</td>
</tr>
<tr>
<td>6109, 653, -0.6785714285714286, -0.22222698298272936</td>
</tr>
<tr>
<td>6114, 652, -0.34523809523809523, 0.2967864982103425</td>
</tr>
<tr>
<td>6127, 655, -0.6785714285714286, -0.2988625666335934</td>
</tr>
<tr>
<td>6139, 654, -0.6785714285714286, -0.3558188959242687</td>
</tr>
<tr>
<td>6150, 657, -0.6785714285714286, -0.4042317758213427</td>
</tr>
<tr>
<td>6150, 656, -0.6785714285714286, -0.4453827237385557</td>
</tr>
<tr>
<td>6170, 658, -0.34523809523809523, -0.48036102945949155</td>
</tr>
</tbody>
</table>

**Figure 4.8:** Extract of sample output used to determine reward: every time an agent calculates reward (when an activity is finished), it outputs its reward and reference reward values to its own personal file. “Time” is in simulation time; “EvalNum” is the count of the number of evaluations (activities) the agent has completed.

the plan has been chosen in the given context. This data can be used determine behaviour, i.e. how many times a particular plan was chosen over the last time period and over different contexts. Considering the number of choices of a plan over a fixed time step allows us to compare the actions most frequently chosen by the characters during that time period.

### 4.2.1 Updating Somatic Marker Preferences

The update of somatic marker preferences (see Section 3.2.2.3, page 94) is achieved by adding to the previous value of the preference. When a new plan-context pair, \(< a_i, s >\), is found, an initial somatic marker preference, \(p_{s,0}(a_i)\), must be assigned. We used a neutral preference, \(p_{s,0}(a_i) = 0\), which means the character does not find the plan either desirable or undesirable. Due to this, the character has no initial, or even random, preference towards or away from any plan. Consequently, as a result of this, all preferences the character generates will be entirely due to its personal experience and not due to random initialisation.

Results from preliminary testing showed that all the characters had a clear preference against insulting others, even when soft goals were manipulated so that the characters should have a preference towards insults. After a more thorough investigation, it was found that the equation used to update the somatic marker preference was
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

Figure 4.9: Extract of sample somatic marker output used to determine behaviour: for a particular plan and context, the agent stores the number of times it has been updated and its current somatic marker preference.

causing these problems. In Chapter 3, the equation to update preferences, Equation 3.7 (page 95) was presented as:

\[ p_{s,t+1}(a_i) = p_{s,t}(a_i) + \beta \cdot d \cdot (r_t - \bar{r}_t) \]  
(4.6)

In this equation, \( d \) is a factor calculated per plan based on the number of paths below this plan (the one whose preference is being updated), \( b \); the number of times this plan could have been chosen in the last activity, \( c \); and the total number of times the plan could have been chosen, \( c_{TOT} \). More exactly,

\[ d = \frac{c}{c_{TOT} \cdot b} \]  
(4.7)

If we examine the goal/plan hierarchy for this domain (Figure 4.2 page 106), let us consider the three top-level activities (although this analysis works for any level in the goal/plan hierarchy) “Move”, “Insult” and “Wait”. These activities are at the same distance from the goal at the top of the hierarchy. However, there are a varying number of ways that each of these plans can be implemented due to the related sub-goals and sub-paths. As a result, the number of paths below each plan, \( b \), is not the same.

For the top-level activities, the number of paths below each plan is: for “Wait” \( b = 1 \); for “Move” \( b = 6 \); and for “Insult” \( b = 15 \). Consider the circumstance where
the character has chosen these activities once each and has calculated the exact same
personal reward, \( r_t \), after completion of each activity. Let us say that the personal
reward is a very high value, meaning that the character should increase its preference
for choosing that activity. However, although the reward was the same, the increase to
the preference for “Insult” will be much less than the increase for “Wait”, due to the
factor of \( \frac{1}{b} \). If this situation is repeated, then, after time, the preference for “Wait” will
be sufficiently more than “Insult” so that “Insult” will not have reached the “desirable”
bucket used during appraisal of choices. The reason for including this factor was to
account for the fact that, although the chosen path of “Insult” was “good”, there are
many other paths that could have been taken that may not have been as good. This was
included to encourage the character to explore all the possible paths before deciding
that the top-level activity itself is “good”. However, given the example just described
and results from the implementation, it was found that “Wait” was being given an
unfair bias towards being chosen. The factor \( \frac{1}{b} \) is a pessimistic way of updating plans.
For a top-level plan to be “good”, all the plans below do not need to be good, we
should only be interested if there is at least one good path.

One way to solve this could be, during the design process, to make sure that plans in
the goal/plan hierarchy all have the same number of possible paths underneath them,
i.e. create a symmetric goal/plan hierarchy. This is probably an unreasonable chore for
the designer and also may not make sense within the domain. After all, in our domain,
waiting probably is a simpler task than insulting someone. We wanted to examine
how different personalities developed when there was no inherent bias towards any
one plan or the other, so we removed the factor of \( b \) from the update somatic marker
equation. It should be noted that, within this domain, it was not possible to execute a
plan more than once within an activity, so that the values for \( c \) and \( c_{\text{TOT}} \) were always
cancelled out. The result of this was that the final equation used to update somatic
marker preference was exactly the same as the theoretical equation for reinforcement
comparison according to Sutton & Barto (1998) (see literature survey Section 2.1.4.2,
page 40), repeated here for clarity:

\[
p_{t+1}(a_t) = p_t(a_t) + \beta(r_t - \bar{r}_t)
\]  

(4.8)
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

4.2.2 Emotionality

Emotionality is one of the three components of the personality template (see Section 3.1.3, page 78). The goal/plan hierarchy was fixed for all characters and implemented as shown in Figure 4.2 (page 106). Preliminary testing of the final domain, where we changed values of both emotionality and soft goal personality, showed that soft goal personality had the greatest effect on behaviour of characters. We wanted to minimise differences in the personality template to determine whether characters could be different without handcrafting both emotionality and soft goal personality. So we chose to fix emotionality. The values chosen allowed for the most stable and “realistic” characters based on preliminary testing. In theoretical emotionality (see Section 3.1.3.3, page 81), the positive and negative values can be different. In our implemented game, we set the values to be the same number to limit handcrafting. The final emotionality used was:

- **Positive and Negative Uptake**: $u = 0.7$. Used when generating an emotion event after a personal reward calculation (see Section 3.2.2.4, page 96). If the personal reward was $r_t = 0.9$ (where 1 is the highest value), an emotion event would be generated with an initial intensity of $I = 0.7 \times 0.9 = 0.63$. This value moderates events so that the characters did not feel them so strongly and therefore their reactions were more smooth and less “jumpy”.

- **Positive and Negative Decay**: $b = 0.003$. Used to determine how quickly an emotion event is forgotten (see Equation 3.10, page 96). This value was chosen based on extensive experimental tests to make sure that events were remembered for a reasonable amount of time for the simulation.

- **Domain-dependent thresholds and soft goal achievement thresholds**: Positive and Negative Threshold set to 0.3. Used to bucket the domain-dependent opinions (attraction to and happiness) and soft goal achievement levels into “H”, “M” or “L”. If the opinion or achievement level is greater than 0.3 of the difference to from the mid-level to the maximum value, then it is “H”, similarly for “L”. That is:
  - if $value > mid + 0.3(max - mid)$, set to “H”;
  - if $value < mid - 0.3(mid - min)$, set to “L”;
  - else, set to “M”.

122
For example, if the range of the value is $(0, 1)$, then the label is “H” when it is above $0.5 + 0.3 \times 0.5 = 0.65$. The label is “L” when the value is less than $0.5 - 0.3 \times 0.5 = 0.35$ and otherwise the label is “M”.

- **Bucketing thresholds**: Positive and Negative Desirable Range $\delta = 0.8$. Used during appraisal of choices when determining which bucket a plan should be in based on its somatic marker preference, see page 88. The plan is considered “desirable” if its somatic marker preference, $p$, is greater than $\bar{p} + 0.8 \times \sigma$, where $\bar{p}$ is the mean preference and $\sigma$ is the standard deviation for all plans being considered at this decision point. Based on the goal/plan hierarchy used (see Figure 4.2, page 106), the maximum number of plans the character was choosing between was three.

- **Bucketing choice threshold**: $\tau = 0.4$. Used in appraisal of choices when determining which bucket to use first, see page 89. A random number, $\epsilon$ is chosen and when this value is greater than $0.4$, then the “desirable” group will be used first (i.e. 60% of the time). When $\epsilon \leq 0.4 b$ (i.e. $b \leq 0.1$), the “undesirable” group will be used first (i.e. 10% of the time). Otherwise, the “neither” group will be used (i.e. 30% of the time).

- **Step-wise learning parameters**:
  - $\beta = 0.2$. Used for update of somatic marker preferences in evaluation (see Section 3.2.2.3, page 94).
  - $\alpha = 0.15$. Used for update of reference reward in evaluation (see Section 3.2.2.5, page 97).
  - initial reference reward, $\bar{r}_0 = 0.5$. Used to initialise reference reward when updating the value (see Section 3.2.2.5, page 97). Preliminary testing used $\bar{r}_0 = 1$ to encourage exploration. However, this made initial personal rewards very low and characters became so negative they were unable to make friends or do anything other than “wait”.

### 4.2.3 Measures of Effectiveness

In order to measure the research sub-questions in relation to the criteria for success (see Table 1.1, page 16) and determine the success of the theoretical model to generate individual characters using adaptation and context, we used three measures of effectiveness: observed behaviour, reward, and individuality. Behaviour is based on the
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

actions characters take in the environment over regular output time periods and reflect the personalities of characters allowing us to test research sub-questions 1a, 1b, 2a, and 3a. The second measure, personal reward, enables us to examine how adaptation and context contribute to the overall performance of the characters, that is, research sub-questions 1c, 1d and 2b. The third measure attempts to find a quantitative value for “individuality”, also relating to research sub-questions 1d, 2b and 3b.

In this section, we examine each of the measures of effectiveness in turn. We discuss how we obtained values from our game implementation and, for individuality, we explain the choice of the particular quantitative measure to be used.

4.2.3.1 Behaviour as a Measure of Effectiveness

Behaviour provides the means for a (human) player to see the personality of the character. That is, behaviour is the player’s view of the characters’ somatic marker preferences and starting personality template. Behaviour is the number of plans executed by characters over a time period that give rise to visible actions in the world. For example, in the screenshot in Figure 4.3 (page 107), we can see that Anna and Chloe are insulting someone, Fran is replying to an insult and since this is a still image, we are unable to determine whether Bec, Deb, and Elle are either moving or waiting. If we watched the game for some time we would be able to count that Anna chooses to insult people very often whereas Fran often waits around and moves. The plans that characters can execute are shown in the goal/plan hierarchy that all characters use (see Figure 4.2, page 106). The top-level activities are moving, waiting, and insulting. Finer distinctions can be made by considering where a character is moving to, where they are waiting, who they are talking to and what they are saying. In our implementation, we output data at fixed time points. From this data (see example extract of data in Figure 4.9, page 120), we can calculate the number of times each plan was chosen over the last time period. We are also able to determine the context used by each character when choosing and executing a plan. This count of action choices over a time periods represents a character’s behaviour.

4.2.3.2 Personal Reward as a Measure of Effectiveness

After every completed activity, a character performs an evaluation of its goals and determines a personal reward value (see Section 3.2.2 page 90). This value reflects
4.2 Experimental Setup and Considerations

how close a character is to achieving all its soft goals, with higher weight placed on those goals it considers most important, according to its soft goal personality. The range of the reward in this domain is from -1 to +1. If the characters are working well within the environment, they should be achieving their goals and obtaining rewards close to +1.

The theory of reinforcement learning (RL) (see literature survey Section 2.1.4.2, page 38) has been applied to both games (e.g., Ponsen et al., 2006a) (see Section 2.2.1.2, page 45) and intelligent virtual agents (e.g., Gadanho, 2003; Sanchez et al., 2004; Seif El-Nasr et al., 1999) (see Section 2.2.2.2, page 56). In these applications, success is usually based on the reward values that the character receives from the external trainer. Another measure of the effectiveness of the model is speed of convergence to the character’s optimal reward value and the stability of reward. Although our model uses an internal trainer, rather than an external trainer, it is still appropriate to consider how the reward for each character fluctuates over time as a measure of effectiveness of our model.

Although the personal reward value appears very functional and may not on the surface improve player immersion, it performs an important role. If characters are not achieving their goals, they will probably look fairly “stupid” to a player. That is, observed actions of the character will not appear to achieve or be related to the functional goals of the games. For example, in the village domain, a character might make bread and then neither sell it nor give it away. This would not contribute achievement of either of the soft goals “have friends” or “have money”.

Further, if a character is not obtaining high personal reward values, then its actions will not be in line with its soft goal personality. That is, the character’s soft goal may be “have money”, but then it gives everything away. If this happens, the character loses credibility and it becomes very difficult for a game designer to create the characters they want. The soft goal personality needs to be followed as part of the personality template as created by a designer. For instance, the designer may want a village full of generous characters who are not trying to “have money”. If these characters are not achieving a soft goal of “don’t have money”, then the characters are not exhibiting the characteristics desired by the designer.
Problems Affecting Reward  When considering the personal reward value, we need to consider the factors that may affect the character’s ability to calculate reward. When a character receives an interaction request from another character, they respond immediately and perform an evaluation on the choices they make, even if they are doing something else at the same time (i.e. the character can listen while doing something else). This may hinder the character’s learning since the personal reward may be based on actions done for the other activity the character is doing at the same time. It is hoped that, over time, the characters will be able to determine the difference based on extended interaction with the simulation.

The game environment is non-deterministic, so when a character performs a particular plan, it may not always receive the same reward. For example, talking to a neutral person may allow the character to find a new friend but not always, because this depends on the other character as well. Further, the character’s chosen plan may not be successful, i.e. may not achieve the hard goal it was supposed to achieve. For example, let us say that Deb wants to move away from her enemy, Elle. Deb places a request with the GUI-side of the simulation. The simulation will find out where Elle is currently and then move Deb to a specific position far away from there. However, while Deb is moving, Elle may also move, so although Deb may arrive at her “destination”, she may not have succeeded at moving away from Elle. In such a way, the resulting personal reward value may not reflect the chosen plans; and so the updated somatic marker preferences may not reflect what actually happened.

4.2.3.3  Individuality as a Measure of Effectiveness

We believe that the model developed will allow for individual characters to be generated more automatically. To determine the success of our model, we need to test to determine whether the characters generated are different from each other. In our model, “different” means the actions that the characters have chosen and implemented are not the same. Ideally, this would be measured based on user perception. However, to test the model initially in a number of scenarios to determine their merit prior to user studies, we will build a more simplistic implementation and determine whether even simplistic implementations can generate adequate individual complexities. To accomplish this, a quantitative measure for *individuality*, based on behaviour, is needed.
Defining a quantitative measure of “individuality” is a difficult task and in our literature survey we found no instances of any such quantitative measure. Often, when determining “success” of virtual characters believability, is measured based on questionnaires of human participants (see Section 2.1.1.4, page 28). Tests using human participants can be very time-consuming and yield variable results. Prior to testing with participants, it is important to determine whether our model can create individual characters that can be computationally detected. We expect that it is easier for computational differences in characters to be detected, compared to people being able to distinguish distinct personalities. So if the differences between characters cannot be computationally detected, then it is unlikely that a human participant would be able to notice any differences in the characters.

Therefore, we propose a computational method to measure differences between characters based on a quantitative measure of individuality. This will allow us to determine whether our model begins to achieve the research question: How can personality be implemented so that the same template can be used to create a number of distinct, individual characters, according to their behaviour? A quantitative measure of individuality will allow us to compare values from different runs to determine the effect of soft goal personalities in the personality template. Differences in individuality can be established by comparing behaviour both with and without adaptation and context. The measure will also allow us to test the model itself to identify which techniques are most suited to achieving our goal. In this manner, by the time human participants are involved in testing, we can be much more confident that our model will generate characters with observably distinct personalities and know which starting conditions will most affect results.

We begin by listing preliminary measures that were considered, but found to be inappropriate for individuality. Then, we discuss what it means to be the “same” or “different”, including a case study that highlights the need to use proportional measurements. We conclude by introducing our chosen individuality measure based on paired t-tests.

Inappropriate Measure: Somatic Marker Preferences Instead of measuring behaviour directly, we considered measuring the characters’ somatic marker preferences. However, in a game environment, what a player can see is more important than what
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

the character would “like” to do, but cannot. The player cannot see what the character wanted to do, only what it actually does. In other words, it is more important to look at the actual observable behaviour of the characters, rather than their desired behaviour. So the final measurement used the actual choices the characters made for the three top-level activities over regular time intervals.

Inappropriate Measure: Final Behaviour We considered looking at the actions that each character executed during the final output time step at the end of the simulation. The characters are able to choose between three top-level activities (see goal/plan hierarchy Figure 4.2, page 106). For each plan, we could specify that character $x$ choose a particular activity either “H”, “M” or “L” number of times in the final time period. Therefore, each character would have a behaviour personality such as “HHL”, representing how it chose each of the three activities. In this way, there are 27 different personality types possible for the characters. A measure of how characters are behaving over the final time period at the end of the simulation could show that differences have been generated between characters and that there is a range of divergence between characters. However, the characters change their actions over the entire simulation. So, it is more suitable to consider how characters behave compared to each other at each and every time point, because participants are more likely to compare characters based on longer time periods, rather than what they are doing at the end of the game simulation. Accordingly, the individuality measure was based on character behaviour over the entire simulation.

Inappropriate Measure: Clustering We considered techniques to cluster characters based on their top-level activity choices. In the ideal result, obtaining highest individuality, each character would constitute its own cluster. Based on research of available techniques it was found that, in order to do an exploratory cluster analysis, a minimum of 50 characters would be required. The desired simplicity of our domain would not be possible with 50 characters. Clustering techniques are unreliable when cluster sizes are small and thus not suitable when we would expect a large number of clusters, each containing very few or only one individual. Clustering techniques were also found to be relatively difficult to automate. They rely on human determination of the appropriate cut-off for the number of clusters after examination of the results.
4.2 Experimental Setup and Considerations

Figure 4.10: Two examples of how characters can be the same for one activity: based on frequency of choice of the activity over regular time intervals. Line represents best fit line.

This makes clustering prone to human error. Given all of these factors, clustering was excluded as a possible measure of individuality.

Definition of “Same” Before considering what constitutes different patterns of behaviour, we begin by considering what makes characters seem to be the “same”. If two characters are very similar, it would be expected they would choose a particular activity approximately the same number of times over a period when facing the same situations (but also at the same time intervals). Let us examine this statement in detail. Two ways in which characters can appear the same are shown in Figure 4.10. The graphs represent a measure of how often the character chooses a particular activity during a time period on the scale [-1,+1], where -1 means the activity was hardly chosen at all. Each dot represents a time period, where the x-coordinate is the number of times character A chose the activity, and the y-coordinate is character B’s value. Figure 4.10(a) represents the circumstance where the characters are fluctuating between choosing this particular activity a lot (top right hand corner), or a little (bottom left hand corner). However, the time periods during which the characters are fluctuating their choices are very similar. Over a given time period, both characters chose this activity the same number of times. In Figure 4.10(b), the characters are changing preferences, but they are always choosing similar values, i.e. character A is almost equal to character B.

1These conclusions are based on research carried out specifically for this thesis by Andrew Buelke and Kaye Marion of the RMIT Statistics Department.
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

Figure 4.11: Four examples of how characters can be different for one activity: based on frequency of choice of the activity over regular time intervals. Line represents best fit line.

Definition of “Different”  Now that we have established what is meant by the “same”, we can consider “different” as being the opposite to this. We believe there are at least four major patterns of behaviour that show characters as “different”, as shown in Figure 4.11. Over one time period in the simulation, Anna could choose “move” a lot, while Bec could choose “wait” often (i.e. choose “move” less often than Anna). Later on, this may have swapped over. This would create a difference between characters, even though the average preference for choosing the activities “move” and “wait” would be the same for both characters. The graph for “move” could look like Figure 4.11(a). In this situation, every time period that Anna chooses “move” a lot, Bec chooses it a very small amount. Figure 4.11(b) shows the circumstance where the characters have fluctuating choices, but over time Anna is always choosing the exact opposite to Bec.

In the examples of characters behaving in the “same” way (see Figure 4.10), we can see that the characters have a positive gradient to their line of best fit. It is tempting to assume that when characters are “different” they have a negative gradient. However, Figures 4.11(c) and 4.11(d) also represent characters that are “different”. In Figure
4.2 Experimental Setup and Considerations

character A is consistently not choosing the activity in question and character B is changing their choice over many time periods, without exhibiting a clear preference towards or away from this activity. In Figure 4.11(d), character A is choosing the activity an average amount, whereas character B is fluctuating between choosing it a lot and not choosing it at all.

Of course, if a character’s choices are not spread over the entire range, then there would be grouping in one corner. For example, Anna may consistently be choosing the activity very often, while Bec is hardly be choosing it at all. These characters should be considered different, but if their data points are clustered very close together it will be difficult to determine whether the best fit gradient is positive or negative. Therefore, the gradient of the line of best fit is unlikely to provide an effective mechanism for measuring difference.

Using Proportions Rather than looking simply at the number of times an activity is chosen, we examine whether proportions offer a useful approach. Let us consider a time period and look at the number of times three different characters chose each of the activities during over the interval:

<table>
<thead>
<tr>
<th>Name</th>
<th>Move</th>
<th>Wait</th>
<th>Insult</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Bec</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Chloe</td>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

In above example, if we compare the characters based on each individual activity using these raw counts, we find that for “move” (for example) Anna and Chloe appear more similar to each other than Bec. However, if we consider the percentage or proportion of times characters choose activities, a different picture emerges as shown below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Move</th>
<th>Wait</th>
<th>Insult</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anna</td>
<td>33.3%</td>
<td>33.3%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Bec</td>
<td>33.3%</td>
<td>33.3%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Chloe</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
</tr>
</tbody>
</table>

Anna and Bec are choosing each activity the same proportion of times, i.e 33.3% of the time, and Chloe now appears different from the others.

In our domain, it was considered more important for there to be a difference between the proportion of times an activity was chosen, rather than the just raw number of
times chosen over the interval. That is, the proportions shown in the second table are likely to provide a more effective representation of differences between characters. This is because some characters may finish activities quickly and complete a larger number of activities within the time period. However, a player watching the characters would probably only notice that one character was choosing an activity proportionally more often than another activity or compared to another character. If we look at the proportion of times each activity is chosen as a percentage of total chosen, rather than the absolute number, we only need to consider values for two of the three activities, since the third is no longer an independent variable (it will be 100% minus the other two).

**Preliminary Individuality Tests: Normality and Chi-squared** To begin testing the behaviour data, the first requirement is to check that the data is normal in order that the validity of the results of the statistical tests is not compromised. Once normality is confirmed, we can use a chi-squared test as a preliminary test. This test relates to the research sub-question 3a (see Section 1.1, page 16) to determine whether our model obtains differences between characters over time. A chi-squared test for every character in the game will determine whether the differences between the characters are significant or not. The chi-squared test can determine whether the behaviour of a character is actually independent from that of another. That is, if we take one character’s actual behaviour as what we expect to obtain, the chi-squared test determines whether a different character’s behaviour matches the expectations, as based on the other character. If the results of the chi-squared test are positive, we can say that not all the characters are the same, some of them are “different”. That is, behaviour of one character cannot be used to predict behaviour of another character. If character behaviour does not pass the chi-squared test, then any apparent individuality is purely coincidental.

**Final Quantitative Individuality Measure** Finally, to determine how “different” the characters are, we use a quantitative measure of individuality. Considering two individual characters, we can use a paired t-test to determine whether the differences between their behaviour (proportion of times they chose a specific activity at fixed time intervals) is significant. The paired t-test (rather than the standard t-test) is used to
eliminate differences from sample points, in our consideration this means differences between one time period and another. That is, we want to compare the proportion of times a character chose an activity compared to another character at the same time interval, rather than comparing how the sum total of how characters behave overall. Consider again Figures 4.10(a) and 4.11(a). In both circumstances, the two characters have the same average choice of the activity. However, at each time interval in Figure 4.11(a) they are choosing the exact opposite activities, whereas in Figure 4.10(a) they are choosing the same. The paired t-test allows us to demonstrate the existence of this difference. The test compares each individual time interval to determine how different the characters are during each time period and returns a p-value indicating how likely it is that the characters are different for the entire simulation. If the p-value is less than 0.05, the characters can be considered significantly different.

If we use the paired t-test on each character against each of the other \(n-1\) characters, then the total number of possible combinations is:

\[
\binom{n}{2} = \frac{n!}{2!(n-2)!}
\]  

That is, the number of possible differences between characters for a particular activity is \(C^n_2\). Since we are looking at proportions of different actions chosen, we need look at only two of the three top-level activities and the maximum number of differences possible is \(2 \times C^n_2\). This gives us a quantitative measure of “difference” or “individuality”. If the individuality value is close to the maximum, then that means almost every character is different from every other character in terms of choices for all their activities. We can also consider individuality per character by comparing one character to all the other characters. The maximum individuality measure for a particular character will be \(2 \times (n-1)\). Individuality per character is a measure of how different this character is to all other characters.

### 4.2.4 Cases and Modes Constructed

In this section, we examine how the different soft goal personalities can be used to form test Cases for the model in relation to the research sub-questions. The Cases allow us to test the characters under a number of different starting personality templates to determine whether some starting templates generate more or less observed personalities.
and to determine the effects on behaviour and reward. We begin by outlining three modes for running the model.

4.2.4.1 Modes

The criteria for success (see Section 1.1, page 16) concern how adaptation and context affect behaviour. To test this, three different running modes are needed for the model itself.

To assess whether the adaptation and context components we have built into the model have the intended effect on characters, the operation of the model with these mechanisms is compared. That is, we need to compare operation excluding the mechanisms of adaptation and context. Accordingly, we use the following three modes.

1. ‘Adaptation Off’: in this mode, characters are unable to adapt their somatic marker preferences and therefore use random choice at every decision point.

2. ‘Context Off’: characters are able to adapt/learn, but cannot distinguish the differences between contexts. That is, somatic markers are stored based on plan name only.

3. ‘Normal’: this is full model as described in Chapter 3. Characters are able to adapt and use both plan and context when storing, updating and using somatic marker preferences.

For clarity, we place the modes within a table showing how adaptation and context affect each mode.

<table>
<thead>
<tr>
<th>Context</th>
<th>Adaptation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>On</td>
<td>3. Normal</td>
</tr>
<tr>
<td>Off</td>
<td>Off</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Note that the mode where adaptation is off, but characters can use context, cannot exist. If the characters cannot learn, then context has no function in our model, unless somatic markers are handcrafted. However, we are looking to minimise handcrafting, so this mode is considered to be outside the scope of this thesis.
4.2 Experimental Setup and Considerations

4.2.4.2 Cases

The overall goal of this thesis is to demonstrate that it is possible to generate distinct personalities without handcrafting a template for each and every character. In our model, a personality template includes the goal/plan hierarchy, emotionality and soft goal personality (see Section 3.1.3, page 78). Initial testing indicated that the soft goal personality was likely to generate the greatest number of differences between characters. To reduce handcrafting, we fixed the goal/plan hierarchy (see Figure 4.2, page 106) and emotionality (see Section 4.2.2, page 122).

We considered five test Cases that covered a range of possible soft goal personality template configurations for the eight characters used in testing.

1. Clear Preference Against One Activity.
2. Multiple Ways to Achieve Goals.
3. Conflicting Goals.
5. Different Soft Goal Personalities.

For the eight characters in the domain, each Case provides a different combination of starting soft goal personalities as detailed in Table 4.2. In each of the first four Cases, there is only one soft goal personality template used by all characters, because the main goal of our work is to demonstrate that a small number of personality templates can lead to significant differences between the overall final personalities of characters. So, if all characters have the same personality template and the model is not effective, then all characters will be indistinguishable from each other at the end of the simulation. In the Case 5, there are four templates, with two characters using each template.

For the soft goal personality templates listed in Table 4.2, the characters were given the same domain-dependent default maximum and minimum soft goal achievement values. Also, the weight that the characters placed on each soft goal was equal. That is, each soft goal was considered just as important as every other one. Each Case was run 10 times with a different starting random seed. After ten runs, the results showed many similarities (based on behaviour, reward and individuality) and no further runs were considered necessary.

The Cases were chosen because they were thought to provide a good spread over possible starting templates in the game domain. The main reasons for choosing each
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

<table>
<thead>
<tr>
<th>Case</th>
<th>Soft Goal Personality Templates</th>
<th>Why Use This Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Clear Preference Against One Activity</td>
<td>1 “Don’t be insulted” “Don’t insult people”</td>
<td>To demonstrate that the characters are able to learn <em>not</em> to choose a specific activity.</td>
</tr>
<tr>
<td>2. Multiple Ways to Achieve Goals</td>
<td>1 “Don’t be insulted” “Make friends”</td>
<td>To allow more diversity for the characters, i.e. have different preferred plan paths, while still achieving the same goals.</td>
</tr>
<tr>
<td>3. Conflicting Goals</td>
<td>1 “Don’t be insulted” “Insult people”</td>
<td>To examine how the characters find a preferred path when there is no clear path to follow.</td>
</tr>
<tr>
<td>4. Complex Soft Goal Personality</td>
<td>1 “Don’t be close to enemies” “Be close to friends” “Don’t be insulted” “Insult enemies” “Don’t make enemies” “Make friends”</td>
<td>A more realistic Case where characters have a number of complex goals to achieve at the same time.</td>
</tr>
<tr>
<td>5. Different Soft Goal Personalities</td>
<td>1 Same as Case 1</td>
<td>To increase the scope for more diverse characters.</td>
</tr>
<tr>
<td></td>
<td>2 Same as Case 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 Same as Case 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 Single soft goal, multiple ways to achieve it: “Make friends”</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Cases used with soft goal personality templates: Note: In Cases 1-4, all characters the same soft goal personality template. Case 5 uses four templates, with two characters per template.
4.2 Experimental Setup and Considerations

Case are listed within Table 4.2 itself. The first two Cases were to verify that, for simple Cases, behaviour is as expected. For the other three Cases, further justification is provided below.

**Case 3: Conflicting soft goals**  By achieving one goal, the character is likely to be decreasing the achievement level of the other goal. This means that characters will be forced to find a compromise.

**Case 4: Complex Soft Goal Personality**  Here, the characters have a larger number of goals that they are pursuing. This represents a slightly more realistic case of the kinds of complex characters that one might want to develop for a real game. That is, characters have many goals they to achieve at the same time. The goals themselves are more complex. Instead of just wanting to insult “people”, the characters want to insult only those characters they do not like.

**Case 5: Different Soft Goal Personalities**  In some respects, this Case is a combination of the previous Cases. It is used to determine how different personality types interact with each other within the same simulation. It is also used to see how the number of starting personality templates affects individuality of characters. The characters in this Case are starting from a base where they already have some differences in what they want to achieve.

4.2.5 Expected results

We have now introduced our game domain, explained our measures of effectiveness and detailed the Cases and modes that we will use to test the model and determine results for the criteria for success. Now we consider the results that we expected to achieve across the different Cases and different modes in relation to the criteria for success identified in the introductory chapter (see Table 1.1, page 16). We separate the expected results according to the measures used, beginning with behaviour, then personal reward, followed by the individuality measure.
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

<table>
<thead>
<tr>
<th>Research Sub-question</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a: Behaviour Over Time</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
</tr>
<tr>
<td>1b: Learning A Functional Soft Goal</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Success</td>
<td>N/A</td>
</tr>
<tr>
<td>2a: Behaviour in Different Contexts</td>
<td>Likely Failure</td>
<td>Likely Success</td>
<td>Success</td>
<td>Likely Failure</td>
<td>Likely Success</td>
</tr>
<tr>
<td>3a: Chi-squared Test</td>
<td>Likely Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
</tr>
</tbody>
</table>

Table 4.3: Expected results for criteria for success related to behaviour. Success or fail relates to whether that particular Case is likely to satisfy or fail each criterion for success as detailed in Table 1.1 (page 16).

4.2.5.1 Expected Behaviour

When examining the criteria for success (from Table 1.1 page 16) relating to behaviour, we consider only the mode with the full model working with both adaptation and context (‘normal’ mode). We do not consider the other modes since we are considering whether our model can satisfy the required criteria, not the effect of adaptation and context. We summarise our expected results in Table 4.3. The reasoning behind these expectations is detailed in the next pages. That is, we now consider each of the criteria for success in turn and consider the effect the different Cases can be expected to have on results.

Behaviour Over Time (Research Sub-question 1a) This criterion relates to how behaviour change over time. In all Cases, we expect behaviour to change and adapt over time due to the adaptation process in the model (see Section 3.2 page 82). Let us examine each Case in detail.

1. Clear Preference Against One Activity: characters should learn that the activity they have a preference against is undesirable, and therefore should choose it less often than the other two possible activities.
2. Multiple Ways to Achieve Goals: characters should each have their own preference for particular activities, as seen in that activity being chosen more often.
3. Conflicting Goals: characters may alternate their chosen behaviour back and forth between two activities.
4.2 Experimental Setup and Considerations

4. Complex Soft Goal Personality: characters will choose some plans more frequently and these values will change steadily (rather than erratically).

5. Different Soft Goal Personalities: characters will choose plans based on their soft goal personality templates, so not all characters will have the same preferred plans.

**Learning A Functional Soft Goal (Research Sub-question 1b)** This measure relates to whether the characters are able to learn a functional goal. It relates particularly to the Case 4, complex soft goal personality, where the characters have many soft goals. Two of these soft goals are: “be close to friends” and “don’t be close to enemies”. We can use this Case to see whether the characters can learn which direction to move. For example, if they are moving with respect to a friend, they should move closer and, if they are moving with respect to an enemy, they should move away. We believe that the characters will be able to satisfy this criteria.

**Behaviour in Different Contexts (Research Sub-question 2a)** This research sub-question relates to how one particular character reacts in different contexts. In other words, is the observed behaviour of a character (actions it chooses over the time intervals) different when it perceives a different context? We list how we believe the different Cases will affect this context-aware behaviour.

1. Clear Preference Against One Activity: characters have a clear preference against an activity regardless of the context. This means that behaviour is likely to be similar no matter what context the character is in. This Case probably will not satisfy the criterion.

2. Multiple Ways to Achieve Goals: characters can achieve their goals in multiple ways, so it is completely feasible that a character may find a different solution path.

3. Conflicting Goals: if characters are achieving one of their goals, they will not be achieving their other goal. This means that the context of which goal is currently being achieved most will greatly affect the character’s action choices. This Case is the most likely to succeed at this criterion.

4. Complex Soft Goal Personality: similarly to the conflicting goals Case (Case 3), it is possible that characters will develop different preferences for different
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

contexts. However, due to the number of soft goals it is attempting to achieve simultaneously, the preferences in different contexts may not be distinct enough to meet the test.

5. Different Soft Goal Personalities: the individuals in this Case mostly have the same soft goal personality template as one of the other Cases. So if the other Cases satisfy the criterion, then there is no need to test this Case as well, since it will also satisfy the criterion.

Chi-squared Test (Research Sub-question 3a) This criterion uses the chi-squared test to determine whether the differences between the characters are significant or not. In Case 1 (clear preference against one activity), the characters would be likely to all display similar behaviour, since one activity should not be preferred, they would all prefer one of the two remaining activity choices. This means that Case 1 is the least likely to pass the chi-squared test. All the other Cases, are expected to be able to develop sufficient differences to pass the chi-squared test.

4.2.5.2 Expected Personal Reward

In many traditional reinforcement learning problems (see literature survey Section 2.1.4.2), reward is expected to stabilise on an optimal solution. However, a game is a non-deterministic environment that is continually changing. Characters may never converge or have a stable reward, due to the fluctuating nature of the environment. Let us consider the criteria for success that are based on personal reward: 1c, 1d and 2d. We will use average reward values across the entire simulation time and across the eight characters to determine the performance of these measures.

Reward Compared to Random Choice (Research Sub-question 1c) This criterion tests whether the average reward that characters obtain using the full framework is higher than the reward they would have obtained if they had used random choice for decisions. That is, we are comparing reward for the ‘normal’ and the ‘adaptation off’ modes. All Cases are expected to satisfy this preliminary test. To fail this test would mean that the characters are not learning effectively at all and they would be better off, in terms of achieving their goals, if they used random choice entirely.
4.2 Experimental Setup and Considerations

**Effect of Adaptation and Context on Reward (Research Sub-questions 1d and 2b)** These criteria relate to how adaptation and context affect reward. So we need to consider the effect across both modes and Cases.

Across the different *modes*, it is expected that reward values will be highest when the characters can both adapt and distinguish contexts, i.e. the ‘normal’ mode. This is because the characters will be able to learn highly specialised information enabling them to make better choices based on past experience in that particular context. The reward will be worst when adaptation is turned off, i.e. random choice. The ‘context off’ mode is expected to have higher reward values than ‘adaptation off’. This is because it is expected that, by distinguishing context, the characters will be better able to determine which plan is the most suitable one. We can summarise these statements by ordering the modes based on which we believe we will achieve the highest to lowest reward: ‘normal’; ‘context off’; and ‘adaptation off’.

Across the different *Cases*, it is expected that the highest reward will be for Case 1 where there is a clear preference against one activity, insults. This is likely to be due to the fact that characters can easily learn not to choose the plan ‘insult’ and can therefore achieve their soft goal. Lowest reward values are expected when the characters have conflicting goals (Case 3) and when there are a number of soft goal personality types (Case 5). In Case 3, the characters will not be able to learn any plan that achieves both their goals, so only one goal at a time will be achieved. In Case 5, characters will have difficulty learning due to the environment in which other characters are using different rules to govern their behaviour. We summarise these statements by specifying the expected ordering of Cases from highest to lowest reward:

1. Case 1 (clear preference against one activity).
2. Case 2 (multiple ways to achieve goals) and Case 4 (complex soft goal personality).
3. Case 3 (conflicting goals) and Case 5 (different soft goal personalities).

### 4.2.5.3 Expected Individuality

The quantitative measure of individuality is used to count how many of the characters are different from each other based on paired t-tests (see Section 4.2.3.3, page 126). We use this measure of effectiveness to consider the following criteria for success: 3b, 1d and 2b. Research sub-questions 1d and 2b examine the effect of adaptation and context.
4. IMPLEMENTATION AND EXPERIMENTAL SETUP

on character behaviour. Research sub-question 3b tests to see whether any characters are obtained that are significantly different from the other characters, i.e. individual.

Effect of Adaptation and Context on Individuality (Research Sub-questions 1d and 2b) These research sub-questions consider the effect of adaptation and context on individuality. We examine our expectations for these criteria both across modes and Cases.

Across the different modes, it is expected that the highest individuality value will occur when the characters are able to adapt and observe contexts, i.e. ‘normal’ mode. This result is expected as a consequence of characters having more precise learning and more ways in which they can acquire different preferences, thus leading to more differences in observed behaviour. If the characters were unable to adapt, their choices will effectively be random and therefore it would be expected that no discernable difference would be found across time or across the different characters within the same simulation. We can summarise these statements by ordering the modes based on which we believe we will achieve the highest to lowest individuality: ‘normal’; ‘context off’; and ‘adaptation off’.

Across the different Cases, we expect the highest individuality value in Case 5, where there are different soft goal personality types. This is because the characters are trying to achieve different goals and so should have different preferences for the activities. We expect the next highest individuality value will be for Case 2, where characters have multiple ways to achieve the same goals. This is because there are a number of possible plan paths the characters can use to improve their personal reward and each character will find its own preferred plan path that is likely to be different from that of the other characters. The lowest individuality value is expected for Case 1, where the characters have a clear preference against insults. In this Case, the characters will have a smaller number of plan paths available to them, since one out of the three top-level activities is undesirable. The other two Cases (3 and 4) are expected to have individuality values somewhere in between the other Cases. We summarise these statements by specifying the expected ordering of Cases from highest to lowest individuality value:

1. Case 5 (different soft goal personalities).
2. Case 2 (multiple ways to achieve goals).
3. Case 3 (conflicting goals) and Case 4 (complex soft goal personality).
4.3 Summary of Implementation and Experimental Setup

4. Case 1 (clear preference against one activity).

Individuality per Character (Research Sub-question 3b) Are any individuals obtained? The individuality value can be used to measure the total number of differences between all characters for a run, as was done for sub-questions 1d and 2b. The individuality measure can also be used to determine how different one particular character is from all the other characters, i.e individuality per character. This sub-question relates to the Cases where all characters have the same personality template, i.e. Cases 1-4.

We use the paired t-tests to count the number of significant differences each character has compared to the other characters. The criterion is satisfied if at least one character is different from the majority of the other characters. Case 1, where each character has the same clear preference against one activity, is the least likely to satisfy the test, since the number of ways characters can differ from each other is reduced. All other Cases are likely to satisfy the test.

2. Multiple Ways to Achieve Goals: Likely Success.

4.3 Summary of Implementation and Experimental Setup

In this chapter, we introduced the school children related domain that was implemented and will be used to test the model. We detailed the three measures of effectiveness (behaviour, reward, individuality) that will be used to assess whether the model satisfies the criteria for success. We identified how these measures could be obtained in a quantifiable manner and predicted results we expect to obtain. We are now in a position to analyse the results of our testing of the model.
4. IMPLEMENTATION AND EXPERIMENTAL SETUP
Chapter 5

Results and Discussion

In this chapter, we present results gathered from testing our implemented domain. For ease of reference, we begin by repeating our initial testing-based research sub-questions and the associated criteria for success (see Table 1.1, page 16). If the model satisfies these criteria for our implemented domain, then the model can be said to succeed in achieving the overall goal: to generate distinct personalities with a minimum of handcrafting. Further, the criteria span the different ways that personality can be measured according to Caspi & Roberts (see Section 2.3, page 66).

1. How does adaptation affect character behaviour?

   (a) Does behaviour change over time? Behaviour changes over time.
   (b) Can characters learn about specific, functional goals? When given a functional goal to learn, the majority of characters choose the “correct” action the majority of the time, based on behaviour.
   (c) How does reward change with time? Reward values are on average higher using our model than when random choice is used.
   (d) What happens if adaptation is turned off? Compared to when adaptation is turned off; both individuality and reward are higher when adaptation is used.

2. How does context affect character behaviour?

   (a) Does character behaviour differ in different contexts? For one character’s behaviour, show that in different contexts the action chosen the majority of times is different.
5. RESULTS AND DISCUSSION

(b) What happens if context is turned off? Compared to when context is turned off; both individuality and reward are higher when context is used.

3. How can personality be implemented so that the same template can be used to create a number of distinct, individual characters, according to their behaviour?

(a) Are the behaviours of characters different from each other over time? Character behaviour passes the chi-squared test.
(b) Are any individuals obtained? Based on their individuality, at least one character is different from the majority of the other characters when they are all based on the same template.

In this chapter, we address all of the above criteria for success in sections based on the three measures of effectiveness used to calculate the results. We start addressing the criteria for success by presenting results that relate to character behaviour (particularly while using the full model, ‘normal’ mode). This is followed by personal reward, and finally the quantitative individuality measure. We then present results relating to personal reward and individuality which were measured across all Cases and all modes. After presenting these results, we discuss implications of the results. The final section identifies some extra findings and several interesting results that emerged during testing, particularly in reference to the domain-dependent knowledge (friendships and happiness) used by characters. We finish with a summary of results.

5.1 Behaviour

We will now look at results from the ‘normal’ mode of each Case. We examine specific example runs to investigate in detail how the characters’ actions change over time, how they learn about sub-plans based on their soft goals, how they behave in different contexts; whether they are different, and how individual characters compare to each other. That is, we are addressing the criteria for success (see Table 1.1 page 16) that relate to character behaviour.

5.1.1 Behaviour Over Time (Research Sub-question 1a)

In this section we are testing to determine whether behaviour changes over time. We found that for all Cases the characters’ behaviour did change over time and responded
to their dynamic environment. We examined the number of times actions were chosen for intervals over the entire simulation in ‘normal’ mode runs for every case. We took a single sample run (out of the 10 possible runs) from each Case and graphed each character on a separate graph. The full set of graphs from these sample runs can be found in Appendix A (page 193). Here, we show two graphs for each Case from two different characters. For example, Figure 5.1 shows the graphs for Anna (Figure 5.1(a)) and Deb (Figure 5.1(b)). In these graphs, simulation time is on the x-axis. Each line represents a different top-level activity (“Insult”, “Move” and “Wait”), and the y-axis shows how many times that particular plan was chosen over the data collection time period. That is, between each data output step, we summed the number of times the character choose each activity, and this is the value shown on the y-axis. The graphs illustrate how each character’s most chosen activity changes over time.

Case 1: Clear Preference Against One Activity  The graphs in Figure 5.1 are taken from a single randomly chosen run of Case 1 using the ‘normal’ mode. In this Case, the soft goal personalities all have a clear preference against one activity, insults. We can see in the sample graphs that, after an initial period of learning, both the characters chose “insult” the least frequently. After this time, approximately half of the characters learned that “wait” is the most desirable activity, while the other half believe that “move” is the best activity. We can see that Anna chose “wait” most frequently (Figure 5.1(a)), while Deb chose “move” most frequently (Figure 5.1(b)).

Case 2: Multiple Ways to Achieve Goals  Two character graphs from Case 2 are shown in Figure 5.2. Although some characters did not have a clearly preferred behaviour (such as Deb, Figure 5.2(a)), we also show here one character whose most chosen behaviour stabilised to be “insult” (Gina, Figure 5.2(b)). In this Case, since there were multiple ways to achieve the goals, some characters (such as Deb and others shown in Figure A.2 page 195) never found a most preferred activity. These characters changed their behaviour to match their environment, while other characters found a stable strategy that worked for them.

1This is true for all characters, as can be seen from the full set of graphs for all characters in the appendices, Figure A.1 (page 194).
5. RESULTS AND DISCUSSION

**Figure 5.1:** Sample character graphs for Case 1 of behaviour based on the individual: Action choices for each character for a particular run of Case 1 (Clear Preference Against One Activity) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis. Each line represents a different activity.

**Figure 5.2:** Sample character graphs for Case 2 of behaviour based on the individual: Action choices for each character for a particular run of Case 2 (Multiple Ways to Achieve Goals) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis. Each line represents a different activity.
5.1 Behaviour

Case 3: Conflicting Goals  Sample graphs from Case 3 are shown in Figure 5.3. In this Case, characters are trying to insult others, but not be insulted themselves. The full set of graphs for the characters in Figure A.3 (page 196) shows that Anna, Fran and Gina (also shown here in Figure 5.3(b)) all have a clear and early tendency towards “insult”. All the other characters eventually settled on choosing “insult” most frequently, but they have some uncertainty over longer time periods. This can be seen for Deb in Figure 5.3(a), where she is alternating between “wait” and “insult”.

Case 4: Complex Soft Goal Personality  In Case 4, none of the characters show a clear tendency towards any of the three top-level activities, see Figure 5.4. A character’s behaviour fluctuates while learning which plan will achieve their soft goals best for that particular context in a changing environment. Since character behaviour is dependent on what the other characters choose, it is possible that each individual character cannot work out the best strategy against the other characters because those characters are changing their strategies as well. That is, the character’s fluctuations in behaviour could be due to the fluctuations of other characters.

Case 5: Different Soft Goal Personalities  The behaviour graphs in Figure 5.5 show an example run from Case 5 where characters have different soft goal personalities.

- Anna and Bec (Figure 5.5(a)) have the same soft goal personality as Case 4, that is, they are trying to achieve six soft goals simultaneously. As we found for the
5. RESULTS AND DISCUSSION

Figure 5.4: Sample character graphs for Case 4 of behaviour based on the individual: Action choices for each character for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis. Each line represents a different activity.

Figure 5.5: Sample character graphs for Case 5 of behaviour based on the individual: Action choices for each character for a particular run of Case 5 (Different Soft Goal Personalities) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis.
5.1 Behaviour

characters in Case 4, Bec’s behaviour fluctuates and there is no clear tendency towards any activity.

- Chloe (Figure 5.5(b)) and Deb have the same soft goal personality as in Case 1, a clear preference against one activity, “insults”. Similarly to the characters in Case 1, Chloe has a low tendency towards “insult”, and has a clear tendency towards “move”.

- Elle and Fran (Figure 5.5(c)) have the same soft goal personality as Case 3, conflicting soft goals. Fran chooses “insult” most frequently, which she must believe will allow her to insult others, while minimising being insulted herself.

- Gina (Figure 5.5(d)) and Heidi have only one soft goal: to make friends. Gina has a clear tendency towards “move” for the entire simulation. Heidi fluctuates most frequently between choosing “insult” and “move”.

5.1.1.1 Comparison to Expectations

In Section 4.2.5.1 (page 138), we outlined our expected results in response to the research sub-question 1a relating to behaviour. We now consider how the model performed compared to these expectations:

1. Clear Preference Against One Activity: characters did learn not to choose “insult” as often as the other activities, as was expected.

2. Multiple Ways to Achieve Goals: some characters chose a particular activity more often than the other activities, as was expected. However, many characters did not have a clear tendency towards any action.

3. Conflicting Goals: several characters had a clear tendency towards “insult”. The other characters alternated back and forth between “insult” and another activity, as was expected.

4. Complex Soft Goal Personality: the characters’ plan choices changed so that the characters often chose each of the activities approximately the same number of times. This was not as expected and may mean that characters were unable to develop clear activity preferences.
5. RESULTS AND DISCUSSION

5. Different Soft Goal Personalities: character behaviour did depend on that particular character’s soft goal personality. For example, the characters given soft goal preferences away from “insult”, show a tendency not to choose that plan. This means that, as expected, not all characters had the same most chosen actions.

We expected that the characters would be able to adapt to their environment and change their behaviour. Characters did indeed change their behaviour based on the other characters in their environment.

5.1.2 Learning A Functional Soft Goal (Research Sub-question 1b)

We now examine whether the characters are able to learn more subtle preferences at the sub-activity level, rather than the activity level. In this section we address the research sub-question focussing on whether characters can learn about specific, functional goals. We consider in particular whether, when given a functional goal to learn, the majority of characters choose the “correct” action the majority of the time, based on behaviour. It should be remembered that in the current implementation, we used a bucket choice threshold so that the most desirable activity group was chosen 60% of the time (see emotionality values used in Section 4.2.2, page 122). This means that, even if a character places its highest preference on the correct behaviour, it is likely to choose this correct behaviour only 60% of the time. The bucket choice threshold was intended so that characters would make a suitable trade-off between exploiting a known successful action versus exploring other possible action.

This criterion relates specifically to Case 4, which allows the characters to learn about functional soft goals. In Case 4, the characters had six different soft goals they were pursuing. Two of these soft goals were: “be close to friends” and “don’t be close to enemies”. If the characters were achieving (or learning) these goals, then they should be choosing the actions “move towards friend” and “move away from enemy”. That is, the characters should be able to learn that these two plans directly achieve each of those two soft goals respectively.

Within the goal/plan hierarchy, the characters cannot choose directly between “move towards friend” or “move away from friend”. In the given goal/plan hierarchy (see Figure 4.2, page 106), the characters are able to choose first which direction they want to move and next who they want to move with respect to. For example,
5.1 Behaviour

if Deb has already chosen to “move towards” someone, she can now choose between “move towards a friend”, “move towards an enemy” or “move towards a neutral”. Although, if Deb currently has no enemies (for example), she will be unable to choose the plan “move towards an enemy”, since the plan will not be applicable.

In Figure 5.6 we show the number of times an example character, Anna, chose each plan over the simulation with a graph for each of the categories of ‘friend’, ‘neutral’, or ‘enemy’. Results from all characters for this particular run can be found in Appendix B (page 199). In the graphs each line represents the number of times that Anna chose move towards or move away.

In Figure 5.6(a), we can see that the line representing “move towards friend” is chosen more times than “move away from friend”. This means that Anna has learned how to achieve the soft goal “be close to friends”. This was verified in results that showed that Anna chose “move towards friend” more frequently compared to “move away from friend”. In Figure 5.6(b) Anna appears to have no clear tendency to move towards or away from characters. This is most likely due to the fact that no soft goals relate to these plans, so the character can choose any plan and it will not affect its achievement of soft goals. Figure 5.6(c) shows the plans that relate to moving with respect to an ‘enemy’. In this situation, Anna appears to prefer to “move towards enemy”, which is the opposite of what she should be learning. This means that Anna did not learn about the soft goal “don’t be close to enemies”.

We now examine the results in relation to the specific criterion. For characters to learn this soft goal, the majority of characters should choose the “correct” action the majority of the time (i.e. over 50% of the time). Here, there are two correct actions: “move towards friend” and “move away from enemy”. Across the 10 runs of this Case in ‘normal’ mode, an average of 6.1 ± 1.1 characters chose “move towards friend” the majority of the time (compared to “move away from friend”). In some instances, a few characters did not ever have friends, so they were never able to choose to move with respect to a friend. An average of 5.9 ± 1.2 characters chose “move away from enemy” the majority of the time (compared to “move towards enemy”).

5.1.2.1 Comparison to Expectations

It was expected that the results would satisfy this criterion. To reach a majority, the average number of characters must be greater than 4 characters. Based on the
5. RESULTS AND DISCUSSION

![Graphs showing Anna’s behaviour when moving for different contexts](image)

(a) Move with respect to a ‘friend’  
(b) Move with respect to ‘neutral’  
(c) Move with respect to an ‘enemy’

**Figure 5.6:** Anna’s Behaviour when moving for Case 4: Action choices for Anna for a particular run of Case 4 (Complex Soft Goal Personality) ‘normal’ mode. In each graph, the number of times the character chose each of “move towards” or “move away” over the time interval is shown on the y-axis.

behaviour data presented above, for both possible “correct” actions, the majority of characters did choose the correct action the majority of the time.

When we examined character behaviour over time in Figure 5.4 (page 150), we noticed that the characters did not have clear tendencies towards any of the top-level activities. Despite this apparent lack of learning at the top-level, reaching the criterion has established that the characters can learn appropriately about the lower-level plans based on their specific soft goals.

5.1.3 Behaviour in Different Contexts (Research Sub-question 2a)

This section answers the research sub-question: Does character behaviour differ in different contexts? The criterion for this question is: for one character’s behaviour, show that in different contexts the action chosen the majority of times is different.

To test this, we need to consider each of the Cases and determine whether any of the characters have different behaviour. For each Case and every character from the sample run, we looked at the actions (in particular, activities) chosen and the context the character was in when they chose that activity. We used data counts of the number
5.1 Behaviour

times each particular character was in each context. Based on this data, we choose the top two contexts to examine in detail. For these contexts, we examined whether one of the three top-level activities was chosen the majority of the time. That is, when the character was in a particular context (e.g. “HH”), we examined whether any of the activities (“move”, “insult”, “wait”) were chosen more than 50% of the time. If an activity was chosen the majority of the time, how did this activity compare to the choices in the second most frequently occurring context. If, in the second context, there was no majority for any activity, or the majority activity was different; then it could be said that the character’s behaviour was different in those two contexts. That is, the criterion is met for that Case. It was unnecessary to test Case 5, since this Case includes characteristics that are can be found in the other Cases.

We found that, in all of the sample runs considered, there were a few characters whose behaviour was different in different contexts, i.e the criterion to obtain at least one character was met. To demonstrate these results, we will present sample graphs for four example characters from each of the four sample runs for the first four Cases. For each character, we show two graphs of behaviour in each of its two most used contexts, for example Figure 5.7. These graphs will indicate that for that particular character, the action chosen the majority of times is different.

Case 1: Clear Preference Against One Activity From Table 4.2 (page 136), the soft goals the characters are trying to achieve are:

1. “Don’t be insulted”; and
2. “Don’t insult people”.

This means that, their ideal context is when they are not being insulted, and are not insulting others. In our domain, this is represented by the context: “HL”. Figure 5.7 shows Bec in her two most common contexts: “HH” (Figure 5.7(a)) and the ideal context, “HL” (Figure 5.7(b)). When Bec is in the ideal context (“HL”) we can see that the activity that she chooses the majority of the time is “wait”, see Figure 5.7(b). In Bec’s other most visited context (when she is insulting many people), her favourite activity is “move”, see Figure 5.7(a). We note, that in both contexts, Bec chooses “insult” the least often. This is in line with her soft goal personality that ensures she should have a preference against insults.
5. RESULTS AND DISCUSSION

![Graphs showing character behavior in different contexts](attachment:image.png)

(a) Context HH: Bec is not being insulted and is insulting people a lot

(b) Context HL: Bec is not being insulted, but is hardly insulting anyone herself (ideal context)

Figure 5.7: Sample character behaviour graphs for two contexts in Case 1: Action choices for Bec for two contexts for a particular run of Case 1 (Clear Preference Against One Activity) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis. Note for this Case the ideal context is: “HL”.

Case 2: Multiple Ways to Achieve Goals  From Table 4.2 (page 136), the soft goals the characters are trying to achieve are:

1. “Don’t be insulted”; and
2. “Make friends”.

This means that, in our implementation, the ideal context is “HH”, which represents when the character is not being insulted and has many friends. The two contexts that Chloe was in most frequently were: “HM” and “LM”. The difference between these contexts is how frequently Chloe is being insulted. In both contexts, she has a moderate number of friends. In “HM”, she is not being insulted much. The context “HM” represents the closest Chloe comes to achieving her soft goals most of the time. The graphs in Figure 5.8 show that, in the context “HM”, Chloe chooses “wait” a clear majority of the time (Figure 5.8(a)). In the context “LM”, there is no clear majority. Chloe’s behaviour is different because, in one context, she clearly chooses one plan above others and in the other context she fluctuates with no clear preference.

Case 3: Conflicting Goals  From Table 4.2 (page 136), the soft goals the characters are trying to achieve are:

1. “Don’t be insulted”; and
5.1 Behaviour

Figure 5.8: Sample character behaviour graphs for two contexts in Case 2: Action choices for Chloe for two contexts for a particular run of Case 2 (Clear Preference Against One Activity) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis. Note for this Case the ideal context is: “HH”.

2. “Insult people”.

For this Case, the ideal context is “HH”, which Deb is in very frequently. Her other most frequent context is “LH”, which represents the Case where Deb is being insulted frequently and is insulting others a lot. When Deb is not being insulted frequently (“HH”), she chooses “wait” the majority of the time, see Figure 5.9(a) When Deb is being insulted frequently (“LH”), she chooses “insult” most frequently. That is, Deb has learnt that, when she is being insulted, her best activity is to insult others (perhaps she has found that this stops them insulting her).

Case 4: Complex Soft Goal Personality From Table 4.2 (page 136), the soft goals the characters are trying to achieve are:

1. “Don’t be close to enemies”;
2. “Be close to friends ”;
3. “Don’t be insulted”;
4. “Insult enemies”;  
5. “Don’t make enemies”; and
6. “Make friends”.

(a) Context HM: Chloe is not being insulted and has a moderate number of friends 
(b) Context LM: Chloe is being insulted a lot and has a moderate number of friends
5. RESULTS AND DISCUSSION

(a) Context HH: Deb is not being insulted and (b) Context LH: Deb is being insulted a lot and is insulting people a lot (ideal context) is insulting people a lot.

Figure 5.9: Sample character behaviour graphs for two contexts in Case 3: Action choices for Deb for two contexts for a particular run of Case 3 (Clear Preference Against One Activity) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis. Note for this Case the ideal context is: “HH”.

The alphabetical ordering of these contexts is important because it allows us to interpret the context strings. The ideal context in this Case is: “LHHHLH”. The two contexts that Gina was in most frequently were: “LLHLHM” and “LLLLHM”. These two contexts differ in the amount that Gina is being insulted. In both contexts, Gina is neither close to her enemies nor her friends, is hardly insulting her enemies, has many enemies and a moderate number of friends. In “LLHLHM”, Gina is not being insulted (more ideal), whereas in “LLLLHM”, Gina is being insulted a lot (not ideal). If we examine Figure 5.10, we can notice that for Gina is rarely choosing any activity in either context. When we examined which context Gina was in most, we found that Gina’s context changed so frequently that she hardly had two clear contexts she was in most. When we ran the simulation for longer, we found similar results, that is, the characters continued to change contexts frequently. This is a problem that is addressed in the discussion of results that follows. From the graphs presented here, we can notice that, similarly to Deb in Case 3, Gina learns that, when she is being insulted a lot, she should insult others, see Figure 5.10(b). This behaviour for this context is different from Gina’s behaviour for her other context where she has no clear preference between the activities but perhaps has a slight tendency towards “move”.

158
5.1 Behaviour

(a) Context LLHLHM: Gina is not close to her enemies nor her friends; she is not being insulted; enemies nor her friends; she is being insulted a is hardly insulting her enemies; has many ene- lot; is hardly insulting her enemies; has many mies; and a moderate number of friends enemies; and a moderate number of friends

(b) Context LLLLHM: Gina is not close to her enemies nor her friends; she is being insulted a lot; is hardly insulting her enemies; has many enemies; and a moderate number of friends

Figure 5.10: Sample character behaviour graphs for two contexts in Case 4: Action choices for Gina for two contexts for a particular run of Case 4 (Clear Preference Against One Activity) mode ‘normal’. In each graph, the number of times the character chose each of the three top-level activities is shown on the y-axis. Note for this Case the ideal context is: “LHHHLH”. Also note that the difference between these contexts is the degree to which Gina is being insulted.

5.1.3.1 Comparison to Expectations

We now compare the results to our expectations that were outlined in Section 4.2.5.1 (page 139).

1. Clear Preference Against One Activity: we believed this Case would not satisfy the criterion since there were less ways a character could differ. However, we did find some characters whose behaviour is different in different contexts, so the criterion is reached.

2. Multiple Ways to Achieve Goals: we expected this Case to reach the criterion and it did.

3. Conflicting Goals: as expected, this Case satisfied the test.

4. Complex Soft Goal Personality: we were uncertain whether this Case would succeed, but our results for Gina show that it did.

5. Different Soft Goal Personalities: since the other four Cases succeeded, this Case will satisfy the criterion as well.
5. RESULTS AND DISCUSSION

<table>
<thead>
<tr>
<th>Research Sub-question</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a: Behaviour Over Time</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
</tr>
<tr>
<td>1b: Learning A Functional Soft Goal</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Success</td>
<td>N/A</td>
</tr>
<tr>
<td>2a: Behaviour in Different Contexts</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
</tr>
<tr>
<td>3a: Chi-squared Test</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
<td>Success</td>
</tr>
</tbody>
</table>

Table 5.1: Results of testing Cases for behaviour-based criteria for success: Success or fail relates to results satisfying the criteria as detailed in Table 1.1 (page 16).

5.1.4 Chi-squared Test (Research Sub-question 3a)

The sub-question relates to whether behaviour of characters differs to other characters. The criterion was that character behaviour passes the chi-squared test. The chi-squared test is a standard statistical tool that establishes whether the behaviour of one character can be used to predict behaviour of another character. If the characters pass the chi-squared test, then the p-value confidence in the result should be < 0.05. This criterion relates to the ‘normal’ mode, so we applied the chi-squared test to characters in every run and every Case and found that the test was passed every time.

5.1.4.1 Comparison to Expectations

Our expectations for this criterion (see Section 4.2.5.1 page 140) were that Cases 2-5 would succeed, but Case 1 was only ‘likely’ to satisfy the test. This means our expectations were exceeded because all five Cases satisfied the test.

5.1.5 Summary of Behaviour Results

The results when using behaviour to assess the performance of characters in the model are summarised in Table 5.1. Comparing this to the expected results, Table 4.3 (page 138), it is seen that our model performed better than expected on all behaviour related criteria for success, since no Case failed any test.

5.2 Personal Reward

Every time a character completed an evaluation step, it output its personal reward value at that time. By examining the graphs of reward versus time, we found large
fluctuations in personal reward values. Due to this, we used an average of the personal reward as the test datum and not the final reward obtained. We collated the reward data across all the Cases and modes to generate the results shown in Table 5.2. Each value in the table represents the average of 80 data points (based on 8 characters in 10 runs), where each data point is the average for each character of approximately 1000 reward calculations over the simulation run. Given the large amount of data, normality tests were not considered necessary.

Since these values are averages, we examined whether these values were statistically significantly different ($p < 0.05$) from each other. We found that in the ‘normal’ mode reward average was always significantly higher than in the ‘adaptation off’ mode. The results for statistical significance comparing the modes to each other are in shown Table 5.3. The relevance of this significance data is discussed in Section 5.2.2 (page 162).

We now discuss these results in relation to the criteria for success relating to reward
5. RESULTS AND DISCUSSION

values. We begin by comparing reward values to random choice and then examine the effect of adaptation and context on reward values.

5.2.1 Reward Compared to Random Choice (Research Sub-question 1c)

The criterion for addressing this sub-question is that reward values are on average higher using our model than when random choice is used. Table 5.2 shows that the average reward values from testing across all Cases and all modes. For the mode using our model, ‘normal’ mode, the average reward values are greater than those generated using random choice, ‘adaptation off’ mode, in all Cases tested. Further, the differences were significant in all Cases, meaning that the model has satisfied this test in all Cases.

5.2.1.1 Comparison to Expectations

Our expectations (see Section 4.2.5.2, page 140) were that our model would generate higher average reward values than random choice. So our expectations have been confirmed and the model satisfies this test.

5.2.2 Effect of Adaptation and Context on Reward (Research Sub-questions 1d and 2b)

The criteria for 1d and 2b (as specified from Section 1.1, page 16) are:

- 1d: Compared to when adaptation is turned off, reward is higher when adaptation is used.
- 2b: Compared to when context is turned off; reward is higher when context is used.

We begin by considering and comparing the reward values across the modes and then across Cases. After this we summarise the results from testing in relation to the criteria and compare to what we expected to find.

5.2.2.1 Reward Across Modes

Effect of Adaptation on Average Reward   Based on raw averages in all Cases, the ‘adaptation off’ mode, performed worse than the other two modes. The difference was significant for all Cases except Case 3, where difference between ‘adaptation off’ and
‘context off’ modes was not significant. This means that, in Case 3 (conflicting goals), without the assistance of context, characters would have been better using random choice to make decisions, rather than learn which choice to make. In Case 3, the characters are trying to achieve two goals that are unlikely to be achieved at the same time, “insult people” and “don’t be insulted” and are actually in conflict with each other. In this Case, context is needed to enable the character to learn that, when one goal is not being achieved, it should attempt to achieve the other soft goal. Although the average for ‘normal’ was higher, it was not significantly better than ‘context off’ mode. The average reward value for ‘normal’ was significantly higher than that in ‘adaptation off’ mode.

**Effect of Context on Average Reward** The ‘normal’ mode produced a higher average reward than ‘context off’ in two Cases as shown in Table 5.2, although this difference was only significant in one Case. In the other three Cases, the ‘context off’ mode produced a higher average reward than ‘normal’, although in one Case the difference was not significant.

### 5.2.2.2 Reward Across Cases

Case 1 showed the highest reward values for the two modes ‘context off’ and ‘normal’. This was the Case where there was a clear preference against one activity. This meant that, at the top-level of choosing between activities, the characters had soft goals that directly indicated that one of the activities should not be chosen. Presumably, that meant it was easier for characters to learn which activities and plans were not good and therefore the characters got closer to achieving their soft goals, i.e. not giving or receiving insults.

Cases 2 and 4 with multiple ways to achieve goals and complex soft goals, respectively, showed the lowest reward values. This is probably because characters found it difficult to find a good reward path. With complex soft goals (Case 4), the characters are trying to achieve too many goals at the same time and are unable to determine which choice to make. The problem is hindered by the fact that reward is not guaranteed. So, although choosing an action like “move towards friend” should improve soft goals, it may not always work. While the character is moving towards someone, that someone could move as well and so the character may or may not get closer to their
5. RESULTS AND DISCUSSION

friend, thus changing the reward value so that it may not reflect the intended action. It was hoped that over time they would learn, apparently the environment fluctuated too much for this to happen. The implementation was run for a reasonably long period of time and the preferences of the characters were not changing greatly towards the end of the period, even if their rewards were fluctuating greatly.

5.2.2.3 Summary of Personal Reward Results

Across modes, actual reward values were, ranked from highest to lowest reward value: ‘context off’; ‘normal’; and ‘adaptation off’. In relation to research sub-question 1d comparing the ‘normal’ mode with ‘adaptation off’ mode, the model satisfied the criteria in all Cases. Reward was higher in ‘normal’ mode than when using adaptation. Considering sub-question 2b, comparing the average reward in ‘normal’ mode with ‘context off’ mode, reward was higher in one Case, equal in two Cases; and less in two Cases.

Across Cases, actual average reward values were, ranked from highest to lowest reward value:

1. Case 3 (conflicting goals);
2. Case 1 (clear preference against one activity);
3. Case 5 (different soft goal personalities).
4. Case 2 (multiple ways to achieve goals); and
5. Case 4 (complex soft goal personality).

The average value for Case 1 ‘adaptation off’ is very low compared to ‘context off’ and ‘normal’ modes. This brings the average across modes for Case 1 down so much that it does not produce the highest average reward value.

5.2.2.4 Comparison to Expectations

Across modes it was expected that the order would be (highest to lowest): ‘normal’; ‘context off’; ‘adaptation off’ (see Section 4.2.5.2 page [141]). Actual results showed the average reward for ‘context off’ as being higher. This could be because context does not actually help the characters learn as effectively as was hoped.

Across Cases, we expected Case 1 to obtain the highest reward and Cases 3 and 5 to obtain the lowest reward values. Contrary to expectations, Cases 2 and 4 had
5.3 Individuality

the lowest reward values. This may be because the characters were not able to find an optimal path from their many choices. The characters may have spent too long exploring their domain and the contexts available to them without exploiting paths that looked promising.

It was expected that Cases 3 and 5 would have the lowest reward values but this was not found to be the result. These are the Cases where there are multiple soft goal personalities in the model (Case 5) and where the characters have conflicting goals (Case 3). In the actual results, these Cases had moderate to high levels of reward. In Case 5, the higher than expected values could be because the individual characters found their own high personal reward values and were not as reliant upon others as expected. For Case 3 with conflicting goals, it appears that the characters were able to find a way to achieve their goals relatively well.

5.3 Individuality

The quantitative measure for individuality used was introduced in Section 4.2.3.3 (page 126). Individuality is determined from pair-wise comparisons of the behaviour of two characters at the same time intervals using paired t-tests. To ensure valid derivation of individuality, the data (action counts over time intervals) must first be checked to confirm that it is normally distributed. All the data we obtained was normally distributed.

A chi-squared test is also required to provide a preliminary check of whether there exists any difference between the characters. If a specific run does not pass the chi-squared test, there are no discernable differences between characters. Individuality gives a quantitative value to just how different the characters are, assuming that some difference has already been detected between characters. None of the Cases in ‘adaptation off’ mode passed the chi-squared test. This means that the behaviour in this mode was not unique to a particular character. This makes sense because, in the ‘adaptation off’ modes characters were essentially using random choice to make decisions. This would be expected to result in non-discernable differences between the characters over time. Both ‘context off’ and ‘normal’ modes passed the chi-squared test in all Cases, allowing us to compare their results for individuality.
5. RESULTS AND DISCUSSION

<table>
<thead>
<tr>
<th>Case</th>
<th>Mode</th>
<th>Context</th>
<th>Statistically Different?</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1: Clear Preference Against One Activity</td>
<td>Off</td>
<td>20.0 ± 4.2</td>
<td>No</td>
<td>20.55</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>21.1 ± 4.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2: Multiple Ways to Achieve Goals</td>
<td>Off</td>
<td>41.7 ± 3.4</td>
<td>Yes</td>
<td>39.25</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>36.8 ± 6.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 3: Conflicting Goals</td>
<td>Off</td>
<td>19.3 ± 5.6</td>
<td>Yes</td>
<td>22.90</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>26.5 ± 4.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 4: Complex Soft Goal Personality</td>
<td>Off</td>
<td>36.4 ± 4.6</td>
<td>Yes</td>
<td>25.25</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>14.1 ± 5.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 5: Different Soft Goal Personalities</td>
<td>Off</td>
<td>42.6 ± 3.4</td>
<td>Yes</td>
<td>40.55</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>38.5 ± 5.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Individuality for test Cases: average number of significant differences between individuals for test Cases. Maximum number of differences is 56.

Individuality uses paired t-tests to determine the number of significant differences between the actual action choices that the characters made for the three top-level activities: “move”, “wait”, “insult”. We examined the percentage of times that they choose the plans over an interval since this is more observable to a user than the actual number of times (see Section 4.2.3.3 page 126). The maximum number of significant differences possible is $2 \times C_8^2$ (see Section 4.2.3.3 page 132). Since we are using eight characters, $n = 8$, we can find that the maximum number of difference between our characters is: $2 \times C_8^2 = 56$. If the individuality value obtained for a particular run was 56, it would mean that all eight characters made significantly different choices to every other character for all three top-level activities. Keeping this maximum value in mind we present the results across Cases and modes in Table 5.4 including whether the difference in each Case between the two individuality values are significantly different. These results show that we found the differences between the two modes to be significantly different in all but Case 1 (clear preference against one activity).

We now consider the criteria for success relating to individuality. We begin by considering the effect of adaptation and context on individuality. Then we consider how well our model performs in generating individual characters with behaviour that is very different from others, research sub-question 3b.
5.3 Individuality

5.3.1 Effect of Adaptation and Context on Individuality (Research Sub-questions 1d and 2b)

The criteria for research sub-questions 1d and 2b (as specified from Table 1.1, page 16) are:

- 1d: Compared to when adaptation is turned off, individuality is higher when adaptation is used.
- 2b: Compared to when context is turned off; individuality is higher when context is used.

The ‘adaptation off’ mode did not pass the chi-squared test and therefore we can conclude that individuality is higher when adaptation is used. We now consider the effect of context on individuality and then how individuality varies across Cases. After this we summarise our results for the criteria and compare them to our expected results.

5.3.1.1 Individuality Across Modes

In this section, we consider the effect of context on individuality, i.e. testing research sub-question 2b, using Table 5.4. In Case 1 (clear preference against one activity), neither mode was clearly better or worse. This could be because the characters all learnt not to choose insults in both Cases, as indicated by the high reward values (indeed these two Case-mode combinations had the highest reward). Now, if they have learnt as they were supposed to, then they would eliminate one of the ways they can differ from each other. This means that perhaps the maximum individuality value is limited by the soft goals they are trying to achieve. In order words, the soft goals for this Case effectively reduced the number of ways that characters could demonstrate different behaviour. So, it is possible that both modes reached the limit and therefore are approximately the same.

In Cases 2, 4 and 5, ‘context off’ mode showed significantly more differences between characters than ‘normal’, i.e had better individuality. In Case 2 (multiple ways to achieve the goals), it appears that context did not help the characters choose their own different way to achieve the soft goals. We were unable to determine why this would be so for this Case compared to others. In Cases 4 and 5, some of the characters had a large number of contexts to learn about (all characters in Case 4 and some in Case 5). So perhaps they were not able to learn their preferred path effectively with the extra
5. RESULTS AND DISCUSSION

contexts to learn about. Certainly, the reward values for Case 4 were very low (see Table 5.2, page 161). This inability to learn could be due to fluctuations in calculations of personal reward. This in turn could mean the characters ended up with no clear preference or behaviour and so could not easily be distinguished from each other.

It is interesting to note that Case 4 had a large difference in individuality between the modes (but curiously not between reward values). That is, when characters used contexts to make decisions, they found it exceedingly difficult to learn a clear preferred path, and so characters may have switched between activities (see for example the characters in Figure 5.4, page 150). On the other hand, with adaptation and no contexts, the characters could learn which plan to choose irrespective of the soft goal context, and so learnt their own way to differ from each other.

The individuality for the ‘normal’ mode was significantly higher than in ‘context off’ mode for one Case, where the characters had conflicting goals (Case 3). This indicates that the individuality was greater when using our full model with contexts. In Case 3, characters had conflicting soft goals, so that using context may help the characters work out which soft goal to achieve next, e.g. the one that is failing currently. Using context appears to have enabled the characters to build up their own unique way to achieving soft goals depending on the context. It shows that, in at least one Case, context-aware characters are more individual.

5.3.1.2 Individuality Across Cases

When comparing how the individuality varied across the different Cases, we use the average across the ‘context off’ and ‘normal’ modes. Based on this, we can see in Table 5.4 (page 166) that Case 5 generated the highest individuality. In Case 5, there were four starting soft goal personality templates used. This meant that from the beginning of the scenario the characters already had some differences between each other. So it makes sense that this Case produced the highest individuality. In fact, it would have been more surprising if this Case had not achieved the highest individuality.

Case 2, where characters had many different ways to achieve their soft goals, had the next highest individuality. In this Case, since their were so many possible optimal paths, characters found their own solution path that was different from the other characters, therefore providing greater potential for individuality.
5.3 Individuality

The least significant differences are in Case 1, clear preference against one activity, and Case 3, where the characters had conflicting goals. It was expected that Case 1 would have a small number of differences between characters because the number of optimal plan paths available to the characters was diminished, as one activity (insults) is never desirable. Case 3, conflicting soft goals, had relatively high personal reward values (see Figure 5.2, page 161), so characters were achieving their goals. The low individuality value may be due to characters achieving their goals in the same way as the other characters.

5.3.1.3 Summary

Across modes, actual individuality values were, ranked in order of highest to lowest individuality: ‘context off’; ‘normal’; and ‘adaptation off’.

Across Cases, individuality ranking was from highest to lowest:

1. Case 5 (different soft goal personalities);
2. Case 2 (multiple ways to achieve goals);
3. Case 4 (complex soft goal personality);
4. Case 3 (conflicting goals); and
5. Case 1 (clear preference against one activity);

5.3.1.4 Comparison to Expectations

Across modes we expected the ‘normal’ would have the highest individuality values since characters would have more dimensions in which to differ to other characters. However, it appears when characters could learn about contexts, they were less likely to have stable behaviour and therefore less likely to clearly differ from each other according to the individuality value. Across Cases, the actual ranking of individuality was exactly as expected (see Section 4.2.5.3, page 142).

5.3.2 Individuality per Character (Research Sub-question 3b)

The criterion for this test is: based on their individuality, at least one character is different from the majority of the other characters when they are all based on the same template. That is, we are considering each character’s personal individuality measure, rather than the entire run’s individuality measure.
5. RESULTS AND DISCUSSION

Here, we are investigating individual characters who begin with the same template, i.e. Cases 1-4. In the previous section, we established that the characters are different from each other. However, if the characters are all only slightly different from each other, then perhaps the differences will not be obvious to an observer. If at least one character is significantly different from the majority of the other characters, then it is much more likely that there is at least one observable personality. For example, if the individuality value is 14, then there are 14 differences between the characters. This could mean that most characters are different from one of the characters on two dimensions (i.e. each character has almost 2 significant differences). On the other hand, it could mean that one character is different from every other character, and therefore that particular character has 14 differences. The second type of differences are probably easier for an observer or player to notice.

In order to consider the individuality value for each character, we separated out the paired t-tests for each character. This allowed us to count for each particular character how many differences existed between it and each of the other characters. The maximum is 14 differences, two for each of the 7 other characters. Remember that characters can only differ from each other on two dimensions, since we use the percentage of times each of the three top-level activities can be chosen (see Section 4.2.3.3 page 131). So, for a character to be different from the majority of the other characters, then that character would have more than 7 differences to the other characters. We then count the number of characters whose behaviour is different from the majority, giving us a value out of 8 (since there are eight characters). Table 5.5 shows the results for the average (across runs) of the number of characters whose behaviour is different from that of the majority of the other characters for both modes. The criterion states that at least one character should be different from the majority of other characters. So, to satisfy the criterion, the average number of characters should be greater than or equal to 1.

5.3.2.1 Comparison to Expectations

Our expectations for our model (i.e. ‘normal’ mode) (see Section 4.2.5.3 page 143) were that Case 1 was likely to fail and Cases 2 to 4 were likely to succeed. As seen in Table 5.5 Case 1 did fail the test, since the average was less than 1 character (0.7 ± 0.8), although some specific runs did satisfy the criteria. Cases 2 and 3 in ‘normal’ mode
5.4 Discussion

Based on the results presented, it is possible to draw some conclusions about our theoretical model and consider whether divergences from expectations were due to the implementation or the model itself. We now detail some key problems that were detected and some possible explanations and solutions, the ideal conditions for improving individuality and several interesting results that emerged during testing.

**Number of Soft Goals** In order to permit behaviour that shows more than one dimensional (simplistic personality) a large number of soft goals (probably at least six) were deemed necessary. However, with \( n \) soft goals, for example, the number of contexts that the character can be in is \( 4^n \) (given that each soft goal achievement level can take on one of four values: ‘H’, ‘M’, ‘L’, or ‘-’). This creates a very large number of contexts to be learning about, especially when the reward is non-deterministic. In order to remedy this situation, it seems that the current context should be something other than the soft goals. Some trials were run in which the characters used their current happiness as their context. This resulted in fast learning, but behaviour was too similar and the learning was not situated enough to fulfill the personality requirements of having context-aware behaviour.

### Table 5.5: Average number of characters who are different from the majority: maximum number is 8 meaning that all eight characters are different from the majority of other characters.

<table>
<thead>
<tr>
<th>Case</th>
<th>Context Off</th>
<th>Normal</th>
<th>Similar Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1: Clear Preference Against One Activity</td>
<td>0.7 ± 1.2</td>
<td>0.7 ± 0.8</td>
<td>Yes</td>
</tr>
<tr>
<td>Case 2: Multiple Ways to Achieve Goals</td>
<td>7.7 ± 0.9</td>
<td>6.0 ± 2.2</td>
<td>Yes</td>
</tr>
<tr>
<td>Case 3: Conflicting Goals</td>
<td>1.0 ± 0.9</td>
<td>2.4 ± 1.4</td>
<td>Yes</td>
</tr>
<tr>
<td>Case 4: Complex Soft Goal Personality</td>
<td>6.4 ± 1.8</td>
<td>0.5 ± 0.5</td>
<td>No</td>
</tr>
</tbody>
</table>

Passed the criterion with values of 6.0 ± 2.2 and 2.4 ± 1.4 respectively. Case 4 did not succeed for in ‘normal’ mode (0.5 ± 0.5). However, for the ‘context off’ mode Case 4 did succeed the test well (6.4 ± 1.8). This could be due to the characters in Case 4 ‘normal’ never having clear tendencies towards certain activities.
5. RESULTS AND DISCUSSION

**Context** Ideally, it was thought that the characters would use their soft goals as context so that they would be able to learn that, when the value for soft goal $x$ was low, they would need to do some particular activity to increase the value of that goal. In our implementation, we found that the characters often switched between approximately two contexts during the same time period. This meant that characters were not experiencing the full range of contexts available to them. This could be due to the soft goal achievement level buckets not being complex enough to capture all information. For example, it may be virtually impossible for any character to obtain any more than a “moderate” number of friends, and hence all characters will appear to achieve “make friends” to a level of “M”. The cut-offs for the achievement level buckets are based on the entire range that the achievement level can take and the ideal value of the soft goal. So future work, may consider changing the ideal value within the soft goal personality. Another hindrance to character learning in any context in our implementation is the fact that the reward values were found to fluctuate greatly, based on fluctuations in the environment.

**Relationship Between Reward and Individuality** The results seem to indicate that high individuality can lead to lower personal reward values. For example, Case 4 with ‘context off’ and Case 2 in both modes show very high individuality and yet very low reward. High individuality, such as in Case 4, means the characters are choosing very different activities at the top-level. However, the different activity choice could be because the characters have not learned how to consistently improve their reward values and are still attempting non-optimal plans. If there is one clear “optimal” path for all characters, then they should all discover this and get higher personal reward values. However, if they have all discovered the same optimal path, then the individuality will decrease. To address these issues, it would be necessary to create more paths that are optimal in the design process. The domain we used was relatively simplistic and therefore the number of possible paths was not as deep as would be likely in a commercial game.

Although it is difficult to draw a conclusion from only five Cases, it seems that if the characters are achieving high reward values, they become more similar and individuality decreases. To confirm this, more Cases with different soft goal personality types within the same Case and across all Cases would need to be used.
5.5 Extra Results: Domain-dependent Measures

**Summary** Despite these issues, the model does show some promise. Character behaviour does change over time (see Section 5.1.1, page 146); and characters can learn about functional soft goals (see Section 5.1.2, page 152). Also the characters are able to exhibit different behaviour in different contexts (see Section 5.1.3, page 154). Further, we found that characters did choose activities in their own unique manner, based on the individuality measure used. The model was certainly better than random choice of the activities, both in terms of reward and individuality. Although there was no clear distinction between whether the use of contexts improved personal reward and individuality, this may be a consequence of the problems listed above. With further work, it is envisaged that context would emerge as being suitable for many domains, especially domains more complex than the implementation used for our testing.

5.5 Extra Results: Domain-dependent Measures

The results presented so far relate directly to the testing-based research sub-questions and criteria posed in the introductory chapter and are based on the generic model presented in Chapter 3 (page 69). When implementing the characters in our game domain, the characters needed some domain-dependent knowledge, see Section 4.1.2.2 (page 110). This knowledge was divided into opinions and facts. Facts are aspects of this particular world that the character can perceive and do not have a judgement value attached to them, for example, my location $\text{1}$.

Opinions are facts with a judgement value attached to them, for example, a character may store: ‘I like Bec’. In our domain, the characters stored two types of opinions: attraction towards others, and personal happiness.

Based on attraction towards others, characters were able to build friendships that were relatively stable. Although how friendships are developed is based on domain-dependent equations, whether they wanted to make friends was part of their soft goals and therefore somewhat dependent on the stability of the model. The characters also had a happiness value that represented how close they were to achieving all of their soft goals.

$\text{1}$In some domains, location may need a judgement value attached to it. For example, the character may need to store: ‘the location I am in currently is bad’
5. RESULTS AND DISCUSSION

In this final section of this chapter, we discuss some interesting findings that relate to the opinion-based domain-dependent knowledge. This thesis did not aim to consider the effects of friendship and happiness on our characters. However, these results represent interesting ways that the characters could interact in the social game that was implemented. These results demonstrate the complexity of our model in generating complex characters and indicate possible future avenues of study. The results themselves are presented in Appendix C (page 205) and show graphs of how happiness and friendships fluctuate over time, and friendship networks generated. We now discuss the implications of these extra results.

5.5.1 Happiness

Happiness was found to fluctuate rapidly from very positive to very negative. However, perhaps this is understandable. The characters are rarely achieving their soft goals and if they do suddenly achieve them, it is often for a reason that they seem incapable of learning. Characters are constantly being insulted and have no direct way to stop this behaviour. In fact, it emerged that the potentially best way to avoid being insulted is to insult other people so that the character can make friends, in the hope that these friends will not turn around and insult them. The domain fluctuates because all the characters are all trying to learn what they should be doing. This means that no character can learn a strategy against a particular character because that character may change their strategy as well. However, these problems are problems that we, as humans, face every day. They are also problems that occur in gaming. Human players will not always play consistently and the character needs to be able to adapt to these changes. So, if we can design characters that can cope with constant changes, then they will be able to function better within games. Fluctuating happiness values indicate that the character was changing to match its environment.

Although no players were used for testing, the model (and implementation) are intended for final use in a game for people to play. When watching the game, players are able to observe whether the characters were “happy”, “sad”, or “neutral” based on their smile (for example see the character images in Figure 4.1 on page 105). In our domain, “be happy” could be used as a soft goal based on happiness levels and would then contribute to a character’s context. This soft goal was not used for any of the Cases presented in this thesis. However, if it was used in a future study, the
player would be able to see a factor that relates directly to the character’s context. If players cannot see which context (or at least partial context) a character is in when it makes a decision, then the decisions that the characters make may seem erratic or unrealistic. Using happiness may be a potential way to make character behaviour more understandable to a player.

5.5.2 Friendships

We found that characters formed stable friendships, even in their changing domain. This is promising given the difficulty of making friends in our model. Characters could not directly make friends. For example, let us examine how Bec and Gina could become friends. Bec could tell Gina an insult about Heidi, and Gina may like Bec more as a result, but only if Gina does not already like Heidi. Bec was unable to store information on who Gina liked or did not like. If Gina agrees with whatever Bec says, then Bec may like Gina more. However, neither Bec nor Gina can do something ‘nice’ in order to become friends more quickly. Our implemented game was designed specifically to be difficult to make friends, in order to test the characters and force them to try more ways to make friends and to force them to choose who to be friends with. The (intentional) construction of the domain made it nearly impossible for one character to be friends with everyone; they had to choose. In some cases (such as Case 1), the characters had no soft goals relating to making friends, so it is not surprising that they only made enemies.

It is very promising that, despite these impediments, the characters were still able to form stable and complex friendships. Further, it is interesting that the complex and relatively stable networks that developed can be related to real world friendships. For example, some of the girls were very popular and formed a sort of ‘clique’, other girls were happy to be on their own and still other girls were trying to get into the clique (that is, they liked the members of the girls in the clique, but they were not liked by the clique). In our domain, there were only eight characters and yet there still existed a large number of possible friendship networks that could be generated. These networks are dynamic and can change if a new character or a human player enters the environment. It would probably be interesting to investigate how the friendship networks change over longer time periods. In summary, this domain-dependent game
application shows that complex friendships can be easily generated and allow each character’s experience to differ from other characters.

5.6 Summary

The results from using the criteria to answer the testing-based research sub-questions can be seen in Table 5.6. Adaptation was found to change behaviour over time, allow characters to learn specific soft goals, and, compared to decisions made using random choice (null hypothesis), adaptation improves both reward and individuality. Characters chose different actions in different contexts. Context was not found to improve individuality and reward in all Cases. Characters were found to be different from each other and, in half the Cases, at least one character was different from the majority of the other characters. Happiness and reward were found to fluctuate greatly over time in response to the dynamic environment. Complex friendship networks were developed showing that our simplistic model was able to generate complex effects. By meeting some of the criteria, the model can be seen to addressing our research goals and questions. We consider the greater meaning of these results in the next chapter as we conclude this thesis.
### Research Questions, Testing-based Sub-questions and Criteria for Success

<table>
<thead>
<tr>
<th>Criteria Satisfied?</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>1. How does adaptation affect character behaviour?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.(a) Does behaviour change over time?</td>
<td>Behaviour changes over time.</td>
</tr>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>1.(b) Can characters learn about specific, functional goals?</td>
<td>When given a functional goal to learn, the majority of characters choose the “correct” action the majority of the time, based on behaviour.</td>
</tr>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>1.(c) How does reward change with time?</td>
<td>Reward values are on average higher using our model than when random choice is used.</td>
</tr>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>1.(d) What happens if adaptation is turned off?</td>
<td>Compared to when adaptation is turned off; both individuality and reward are higher when adaptation is used.</td>
</tr>
<tr>
<td>Individuality: Yes</td>
<td>Reward: Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. How does context affect character behaviour?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.(a) Does character behaviour differ in different contexts?</td>
<td>For one character’s behaviour, show that in different contexts the action chosen the majority of times is different.</td>
</tr>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>2.(b) What happens if context is turned off?</td>
<td>Compared to when context is turned off; both individuality and reward are higher when context is used.</td>
</tr>
<tr>
<td>Individuality: Yes in 1/5 Cases</td>
<td>Reward: Yes in 1/5 Cases</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. How can personality be implemented so that the same template can be used to create a number of distinct, individual characters, according to their behaviour?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.(a) Are the behaviours of characters different from each other over time?</td>
<td>Character behaviour passes the chi-squared test.</td>
</tr>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>3.(b) Are any individuals obtained?</td>
<td>Based on their individuality, at least one character is different from the majority of the other characters when they are all based on the same template.</td>
</tr>
<tr>
<td>Yes in 2/4 Cases</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.6**: Summary of results of testing-based research sub-questions based on criteria for success (as introduced in Table 1.1, page 16). Words in *italics* are the measures of effectiveness.
5. RESULTS AND DISCUSSION
Chapter 6

Conclusion

The main aim of our research was to develop a method of generating individual characters with a minimum of handcrafting so as to populate virtual environments with large numbers of interactive characters. We have developed a model that generates individual characters with the same initial personality template. The results showed that the personalities generated were adaptive, characters developed different activity preferences for different contexts, and the characters were distinctly individual (in a computational sense). In this chapter, we begin by discussing the research questions in relation to our final model. Next, we discuss further testing and future work that could improve the model. This is followed by the implications of this model for the games industry and some final words to conclude the thesis.

6.1 Addressing the Research Questions

The major research questions presented in the introduction (see Section 1.2 page 14) relate to building and testing a model of personality so that the characters generated are adaptive, context-aware and individual. We consider each of the research questions in turn and address the related model-based and testing-based sub-questions.

6.1.1 Research Question 1: Adaptation

How can a model of personality be created that uses adaptation? How does adaptation affect character behaviour? Personality can be seen in the actions taken by characters. Characters choose which actions to execute based on their
preferences (implemented as somatic markers). So, to include adaptation in personality, the model needs to adapt the preferences. This can be done by updating the preferences after execution of actions within a feedback loop. Self-reinforcement (personal reward) learning (from cognitive-social theories) can be used so that the preferences learnt are based on the personal soft goals of the character. Which soft goals a character is trying to achieve (and the importance of each soft goal) is part of the character’s personality template. So the personality template affects how the character learns the adaptive part of its overall personality. We now consider the model and testing-based sub-questions relating to adaptation.

- Model-based sub-questions:
  - **What aspects of personality can adapt?** Somatic marker preferences, which are used to make decisions, can change based on experience and are part of an character’s personality.
  - **How are decisions made?** Decisions are made by following the appraisal of coping choices process (inspired by the somatic marker hypothesis) that uses somatic markers to place preferences on available plans in the current context.
  - **How can characters calculate reward?** Reward can be calculated based on a weighted sum of an individual’s current achievement levels of its soft goals (based on its soft goal personality).
  - **How can characters use reward to change behaviour?** Personal reward can update somatic markers according to the reinforcement comparison technique. Somatic markers affect behaviour by changing the character’s preferences towards or away from choosing certain actions.

- Testing-based sub-questions:
  1.(a) **Does behaviour change over time?** According to our results, the characters were able to change which actions they executed most frequently based on feedback from the environment and their personal goals. This means that characters can change to suit different players or design changes to the environment. We met this criterion because the character’s somatic markers adapted over time, so that the preferences for action changed, based on personal reward.
6.1 Addressing the Research Questions

1.(b) **Can characters learn about specific, functional goals?** Our results showed that characters could. This means that a designer can give a character a goal to achieve in the world or a role to play in the game and the character will be able to automatically learn how to do this (i.e. without further external intervention). Further, it means that the overall character personality that is generated can still fulfil functional requirements. We met this criterion because characters learn based on personal reward and personal reward is based on the soft goals the character is trying to achieve.

1.(c) **How does reward change with time?** Reward fluctuates greatly in response to fluctuating environmental feedback. The reward values that the character calculates are higher over time when the character can use adaptation. This means that adaptation improves achievement of soft goals.

1.(d) **What happens if adaptation is turned off?** The personal reward and the individuality of characters decreases. In terms of reward, this means that adaptation helps characters learn how to achieve their soft goals, which improves their reward. In terms of individuality, the results indicate that characters are more different from each other when they can adapt. Characters who are choosing actions randomly may appear different over one time period but, over an entire game, these characters will not be significantly different from each other. Characters who can learn their own preferred actions to achieve their soft goals will become different from the other characters over the entire game.

6.1.2 Research Question 2: Context

**How can a model of personality be created that uses context? How does context affect character behaviour?** Characters execute different actions depending on their context. Characters choose which actions to execute based on their preferences (somatic markers). So, the character’s preferences need to be based on context as well. In our model, we choose context to relate to the soft goals the character is trying to achieve (part of its personality template). To know what to do in a specific context, the character learns, via adaptation, their preference for that particular action and context (rather than hard-coding the preference). We now consider the model and testing-based sub-questions relating to context.
6. CONCLUSION

- Model-based sub-questions:
  - How can context be represented? Context is a string representing a discrete version of the achievement level of each of a character’s soft goals from its soft goal personality.
  - How can context information be provided? Characters update their somatic markers based on the context they were in when they made the decision they are evaluating.

- Testing-based sub-questions:
  2.(a) Does character behaviour differ in different contexts? We found that the most preferred action was different for a character in their two most frequently used contexts. This means that characters can learn which action they personally consider most suitable according to whether specific soft goals are or are not currently being achieved. This results in more complex and realistic behaviour and therefore overall personality. We met this criterion because, when characters learn somatic markers, they link their reward to the action and the context in which the action is chosen.
  2.(b) What happens if context is turned off? When context is turned off, reward and individuality are likely to improve, according to our results. This is the opposite of what was expected. In terms of individuality, this means that being able to learn about different contexts did not allow characters to consistently find a clear solution path that is different from the other characters. In terms of reward, this means that characters are less likely to learn how to achieve their soft goals when they have many soft goals to achieve simultaneously. This could be because there were too many contexts in some Cases or the way contexts operated within the specific implemented domain.

6.1.3 Research Question 3: Individuality

How can personality be implemented so that the same template can be used to create a number of distinct, individual characters, according to their behaviour? Personality is made up of a fixed template and an aspect based on individual personal experience. Different personal experiences will result in different
preferences for actions, and therefore different actions executed, leading to different observed personality, even when the template is the same. We now consider the model and testing-based sub-questions relating to individuals.

- Model-based sub-questions:
  - **What is an individual within our model?** An individual is comprised of a number of components and beliefs including personality template, somatic markers and domain-dependent knowledge. Individuality is distinguished based on observable behaviour, in particular the actions that a character executes over a time period.
  - **What is a personality template?** A personality template guides the character’s learning and decision-making. In our model, it is made up of a goal/plan hierarchy, a soft goal personality and emotionality.
  - **How does personality change over time (i.e. how can a character be different from another character with the same template)?** Personality changes over time by adapting somatic marker preferences. Somatic markers are adapted based on personal reward experience, so no two characters will have exactly the same somatic markers, even if they began with the same initial personality template.

- Testing-based sub-questions:

  3.(a) **Are the behaviours of characters different from each other over time?** Characters were different at the level of the actions they executed. Results showed that characters do not have the same most frequently chosen actions as other characters over an entire run. We found that the behaviour of one character cannot be used to predict the behaviour of another over a run. We obtained this because the learnt somatic markers allowed the character to find stable ways to achieve their goals in a different way compared to other characters.

  3.(b) **Are any individuals obtained?** We found that in half the Cases, we were able to obtain at least one character that was significantly different from the majority of the other characters. This means that some characters are more different from others and are therefore more likely to have an observably different personality. We obtained this result because characters had their
own unique experiences, which changed the somatic markers they obtained, which in turn changed the actions that character preferred to execute.

6.1.4 Implications Arising from Research Questions

Our results indicate that our model can achieve our aim to build individual, adaptive and context-aware characters from the same template. This means that individual preferences for every character for every context do not need to be hard-coded, they can be learnt based on an automatic processes and their personality template. Traditionally in personality research for virtual characters, it is very difficult to quantify results. We have used a set of criteria for success for our model to satisfy prior to user feedback questionnaires and studies. These criteria allow us to determine the areas of our model that are in need of improvement, in this circumstance, context-aware behaviour. That is, the benefits of context were not as clear as was expected. Satisfying the criteria does not mean that the model can necessarily produce characters that are distinct to human observers. However, satisfying many of the criteria indicates that the model has promise. The model did not fully satisfy all criteria, so further testing is needed. In future testing, it might be possible to determine a value above which the individuality measure should be in order for characters to be observably different according to players. That is, players could be presented with several versions of the same game, but with different character individuality measures, and, based on feedback, a minimum individuality value may be discovered. The implemented domain was relatively simple domain, yet it could generate complex friendship networks and obtain distinct differences between the characters. The results showed that adaptation allowed characters to learn their own unique way of achieving their personal goals. However, there is scope for improvement in the level of individuality and the spread of differences across characters, particularly compared to when context is not used.

6.2 Potential for Future Research

The model presented in this thesis shows some promise as a method for generating many complex personalities from a single personality template. However, the results indicate that further work is needed to resolve some of the problems that emerged during testing, particularly relating to context and improving individuality per character. We
begin by discussing tests that could be done on the model as it is now relating to personality templates and input parameters. Next, we investigate how the model could be improved or extended, based on current results. Then, we consider visualisation of some of the complex components from our implementation. Finally, we discuss several different qualitative tests with humans that could be performed to test the model (or future models) further.

6.2.1 Personality Templates and Input Parameters

For our testing, we choose a small variety of Cases to demonstrate that the basic principles of our model were effective. When choosing the Cases, we fixed many parameters so that we could use simple Cases to test the model extensively for those Cases. However, there will always be more Cases and further tests that could be completed on the current implementation of the model. By investigating the current model and implementation with more Cases, we may be able to establish more precisely the effect of context on reward and individuality and whether high individuality always leads to low reward values. This process will allow us to pinpoint which areas of the model or implementation need more work, and which areas are already suitable.

Although we have validated the model against the criteria for success relating to our testing-based sub-questions, more stringent levels for the criteria could allow further benefits and problems to emerge. Characters had the same soft goal personality, meaning they had the same weights and ideal values for the soft goals to be pursued. It would be interesting to compare Cases where the soft goals were the same but the weights and/or ideal values were different for each character. Part of emotionality is the trade-off between exploration and exploitation, which was not fully investigated in the implemented domain. A more thorough investigation into the influence of the learning parameters ($\alpha$, $\beta$ and bucketing-related values) may give more stable learning and less abrupt changes in preferences. One possibility would be to change this parameter over time so that initial values encourage the character to explore its world and later encourage the character to stay with its preferred choices. The results from the current Cases indicate that there was a trade-off between individuality and personal reward. That is, some Cases that showed high individuality also had characters generating low personal reward values. This could have been a side-effect of the specific Cases chosen so more Cases would need to be constructed to resolve this. With different Cases it
could be necessary to investigate extending runtime and changing output time steps further.

6.2.1.1 Testing Adaptation Further

In our testing, we compared characters using our model in modes where adaptation was turned off. Turning off adaptation meant that the characters were effectively using random choice to make decisions. In order to further test the effect of adaptation, a method would be to run the simulation for a specified duration. After this, character data could be collected, that is, the character’s somatic marker preferences that were developed. Then, the simulation should be run again twice: firstly where the characters can continue to adapt, and secondly where the characters cannot change their somatic marker preferences any further. We could then test each run against the criteria. This test would allow us to determine whether the benefit from adaptation is only during the development of character personality, or whether continued adaptation improves or stabilises behaviour, reward and individuality.

6.2.1.2 Designing Specific Characters

Although high individuality was one of our goals, if there is too much individuality within the same personality template the designer will lose control over the types of characters generated. There is obviously a trade-off between the amount of handcrafting of a character and the predictability of their behaviour. On the one hand, if a character is required to fulfil a specific, high-profile, role within the game, then its behaviour and personality probably need handcrafting. On the other hand, for the background characters who contribute to the overall feel of the world, high diversity or individuality with a minimum of time-consuming handcrafting would be more desirable. However, even with background characters a designer does not want them acting in entirely unpredictable ways. That is, a designer may want a number of characters who are “greedy” and some who are “friendly”. In our model, these personality types can be set up using a different personality template for each major group. We have shown from our results that, within the same personality template, a number of different characters can be generated based on their preferences for different activities. We need to investigate further whether, after time, the range of characters generated still fit within the overall personality template they were given. That is, if a number of
6.2 Potential for Future Research

characters are given a “greedy” personality template, then we need to check whether all the eventual personalities generated remain reflective of how a “greedy” character should act. Although testing would be needed for confirmation, given the way that the soft goal personality has been implemented to constrain characters to achieve specific goals, it is not envisioned that there would be any significant problems in relation to this question.

6.2.2 Model

When designing the model and testing it, a number of ideas emerged as worthy of further investigation. We considered how a somatic marker should be updated if the chosen plan or activity failed. For example, the character wants to talk to someone, but is unable to find someone to talk to. Initially, we decreased the preference value for such failed plans. This led to many plans having a negative preference and the characters did not want to choose any of the plans. In the final implementation, there was no penalty for failing to complete execution of a plan. However, it seems that the characters should be able to learn which plans are difficult to execute and that there should be a better method of adjusting the somatic marker value to reflect this. One possible method would be the redistribution of reward system proposed by Ponsen et al. (2006a) (see literature survey Section 2.2.1.3, page 46).

The results show that adaptation in our model improves personal reward and individuality but indicate that the use of context does not consistently improve either of these measures. Does this mean that context is only appropriate in certain domains? Or that the manner in which we implemented context was not appropriate? Perhaps context should relate to something other than goal achievement. Certainly, it was clear that even with only six soft goals there were a large number of contexts the character could be in. This means it is very difficult for characters to learn which action to choose in each context. One solution to this could be to determine a similarity measure across contexts, so that characters could use information from similar contexts to make a decision about their current context. One method is that adopted by Bakkes & Spronck (2006) (see literature survey Section 2.2.1.3, page 48). In this method, the characters can use preferences from similar contexts when encountering a new context. Another method is that every time the character makes a decision, it could use a weighted
function to use preferences not just from its current context but also from similar contexts. A final method is to allow characters to learn for some period of time without context, and then use this learning as a baseline for future learning including context information. We found that reward values fluctuated frequently and did not stabilise or converge. This may be desirable in some game applications, but if it is found to be a problem, then techniques to improve convergence of the learning could be considered, such as those presented by Matignon et al. (2006) (see literature survey Section 2.2.2.2, page 57).

6.2.3 Visualisation of Personality, Context and Friendship Networks

The implemented game could generate fairly complex personalities, contexts and friendship networks, despite the simplicity of the actual game structure. However, these three aspects were not directly observable to a player. We now consider some possible games or domains that could exploit further these aspects.

In our implementation, character personality was only visible in the number of times they choose each activity. The characters were only able to choose three obviously different activities and it was questionable whether a player watching a character would be able to distinguish the fact that one character chose a particular activity more frequently than any other activity and, as a result, link this to a different underlying personality. However, it is unclear how to make personality more explicitly obvious. A more in-depth game scenario would give the characters more observably different ways to achieve plans and to allow their personality to be expressed, which would hopefully result in more explicit personalities. The village example from the introduction could be a good starting point.

If contexts are confirmed to be a useful construct for learning personality, there needs to be a better way to enable the player to distinguish that a particular character is in a particular context. The characters were able to smile or frown to indicate their happiness, but this discrete indication cannot convey the complexity obtained when the context is based on many soft goals, such as “LLHLHM” or even “HM”. If the domain leads to characters choosing very different activities in different contexts, then the player must be able to distinguish at least some of the different contexts so that the player can understand (and therefore believe) the behaviour of the character.
6.2 Potential for Future Research

The domain-dependent friendship networks generated are interesting to examine from a real-world perspective to consider how these networks would be generated or function. For example, in some Cases there was a clear clique of popular characters. It would be interesting to design a game, that exploited these friendship networks in a more visible manner, so that players could notice how the friendships change and alter over time. Perhaps the friendship network could play an explicit role in the player’s game, so that the changing network could be monitored and the player could attempt to put themselves at the centre of the network.

6.2.4 Qualitative Testing

In order to determine the success of the model in improving individuality and believability, it will eventually be necessary to use human participants to play the game and answer questionnaires. Although we determined a quantitative measure of individuality, the measure loses significance if the measured differences are unable to be detected by a human participant. The game scenario was chosen to be simplistic, due to time and resource constraints, and also in the hope that personalities would be easier to distinguish in a simple environment. However, to distinguish personalities, a human participant would need to play the game for a long period of time, something that is perhaps undesirable for such a simple game.

Ideally, we would like to measure the coherence of the characters. That is, determine whether their behaviour is stable, identifiable as uniquely theirs and not erratic across time and contexts. It is likely that this can also only be measured using questionnaires of participants. The problem with developing questionnaires is that some participants may be likely to notice differences where there are none and vice versa. To reduce this effect, it would be necessary to ensure that some of the participants are also given a version of the game without our model being utilised fully, i.e. blind tests. In this way, participants could compare characters between the Cases.

Ruttkay, Dormann & Noot (2002) proposed a framework to compare embodied conversational agents (ECAs) to each other and to traditional input methods (see literature survey Section 2.1.1.4, page 28). However, this framework still relied on many subjective questions based on the opinions of individuals. To test for different personalities, we would need to attempt to answer such questions as: “how many different characters did you notice in the game?” What one participant uses to measure
6. CONCLUSION

“different” may be nothing like another participant’s measure. Determining whether the model quantifiably “improves” believability and individuality may not be achievable depending on the robustness of the questionnaires used, the method and the number of participants in the study. We propose another method of determining whether distinct personalities are generated in any simulation: the personality guessing game.

6.2.5 The Personality Guessing Game

The original Turing test proposed by Alan Turing \cite{Turing1950} in 1950 related to distinguishing a woman from a man and further whether a machine could be distinguished from a woman \cite[Section 2.1.1.4, page 30]{text}. Imagine a similar test where you are at a terminal typing to two of your friends. If they change their names, are you able to distinguish the two from each other? If they are good friends, you probably can.

Imagine a game world populated with some fixed number of characters. The interrogator (a player) watches and interacts with the characters for some amount of time to “get to know” who is who and what the characters like to do. That is, the interrogator attempts to observe each character’s personality. After some time, we stop the simulation. We change the names and physical appearance of the characters in some way, but do not change their underlying personality and reasoning. That is, although the characters visibly look different, the control of their behaviour is unchanged. The interrogator now watches the characters again. If the interrogator is able to correctly guess which character matches to which original character name then the test would be said to be satisfied. This test gives the interrogator a quantifiable score for the test. This quantifiable value can then be used to measure the effectiveness of the model.

This personality guessing game represents an ideal test of personality in a simulation since it eliminates much of the subjectivity of standard questionnaires. If the underlying personalities are strong enough, they will be observable even when the virtual appearance of the character has changed. If we performed this test using characters that were all generated from the same personality template, we would be able to demonstrate more strongly that our model does automatically generate individual personalities for each character. That is, if the participants are able to distinguish the individual characters better than by random chance, then we can be confident that the personality of each character is individual.
### 6.3 Implications for Games

The implementation presented in this thesis was of a more academic nature (compared with commercial games), so as to test the model itself. In this section, we consider the model and results obtained and how these pertain more broadly to the games industry. The model developed is generic and can therefore be applied to a large number of different domains, unlike some models that require a complex domain-dependent appraisal of choices. To apply our model to a particular domain, the available personality templates and the equations to determine soft goal achievement levels must be created.

The characters developed using our model are adaptive, context-aware and individual. *Adaptation* improves the number of distinct individual characters, compared to randomising behaviour. Adaptation means that the characters can learn functional goals within the environment, can change automatically to match a new or changing environment, and will automatically learn about the player. The designer does not need to predict all the situations a character will be exposed to because the character can adapt automatically. Furthermore, the character cannot learn entirely new behaviour and become a liability to the game due to its personality template (in particular the goal/plan hierarchy) which limits the behaviour it can exhibit. In this way, characters are not themselves unpredictable, it is the actions they choose to execute that differ.

The instability noticed in reward and character behaviour may be well suited to a dynamic game environments. In a dynamic environment, the character would need to be changing their strategies constantly, something the characters generated in our implementation do without effort by the designer. Some game applications take a personality template and use this combined with a new random factor for every decision. However, our results showed that random behaviour does not give distinct individuals over time. *Context* allows characters to choose behaviour that is appropriate to the situation the character is in. By combining context with adaptation, the designer can allow the characters to learn what to do in every situation. Our results indicate that context may not be as useful as originally expected. Without further research, it may be simpler to remove contexts from characters in a game environment.

*Individual* characters are generated from a single personality template. This means that a designer can handcraft a small number of personality templates or archetypes to be used by a large number of characters. Although characters within an archetype will have some similar behaviour, each will be a distinct individual in their own right.
6. CONCLUSION

The player feels that they are continually meeting new characters, rather than the same character with the same archetype. This adds to the immersion and depth of the virtual world generated. In order to test our model, we developed a quantitative measure of individuality. This measure is useful in itself so that a designer can test character personality prior to obtaining potentially costly and lengthy player feedback via questionnaires and playtests.

Our model is intended for use in games needing many background characters. Since we sought to analyse characters in detail, we limited testing so that only eight different characters were considered for our illustrative Cases. However, in four of our Cases, all eight characters had the same starting template and yet they developed so that there were measurable differences between the characters in all Cases. Our implementation could easily be scaled to generate more characters from the same template since, once a personality template has been designed, only a single number in the input text file needs to be changed so that the implementation instantiates fifty characters of a particular template, compared to eight.

6.4 Final Words

The model we have designed and implemented shows some promise as a method of developing complex character preferences or personality without handcrafting behaviour for each character in every situation. One of the main technical contribution of this research is the inclusion of soft goals and somatic markers in a BDI architecture. As noted in this conclusion, further research is required to improve the implementation domain and contexts within the model, since the domain was found to be too simplistic and contexts did not clearly improve the characters. In the introduction, we presented an ideal game world that would benefit from our model, the village example. At the end of this thesis, we believe that an implementation of the village example would benefit from using our existing model. This thesis has shown that, through using soft goals and somatic markers to allow adaptation and to perceive contexts, we have developed a model that enables personality to influence decision-making and evaluation for characters in a virtual world. It demonstrates that it is possible to automatically generate, from a single personality template, multiple characters that are unique, adaptive and context-aware, without handcrafting all behaviour.
Appendix A

Behaviour Over Time

This appendix contains graphs for each sample run of each Case (for every character) in relation to the testing research sub-question 1a: does behaviour change over time? In other words, do the activities that the characters choose change over time? Each graph shows how a specific character’s behaviour changes over time during the simulation for each Case when using our model. In this instance, we are only considering ‘normal’ option runs, where the model is being used with both adaptation and context.

We take a sample run from each Case and plot each of the eight characters on a separate graph. For example, in Figure A.1 we see the graphs for Anna (Figure A.1(a)) and Bec (Figure A.1(b)) from Case 1. In these graphs, simulation time is on the x-axis. Each line represents a different activity (“Insult”, “Move” and “Wait”), and the y-axis shows how many times that particular plan was chosen over each data collection time interval. The graphs can be used to see how each character’s most chosen activity changes over time.
Figure A.1: Case 1 behaviour based on the individual: Action choices for each agent for a particular run of Case 1 (Clear Preference Against One Activity) option ‘normal’. In each graph, the number of times the agent chose each of the three top level activities is shown on the y-axis. Each line represents a different activity.
Figure A.2: Case 2 behaviour based on the individual: Action choices for each agent for a particular run of Case 2 (Multiple Ways to Achieve Goals) option ‘normal’. In each graph, the number of times the agent chose each of the three top level activities is shown on the y-axis. Each line represents a different activity.
**A. BEHAVIOUR OVER TIME**

![Graphs showing behaviour over time for different agents.](image)

- (a) Anna
- (b) Bec
- (c) Chloe
- (d) Deb
- (e) Elle
- (f) Fran
- (g) Gina
- (h) Heidi

**Figure A.3:** Case 3 behaviour based on the individual: Action choices for each agent for a particular run of Case 3 (Conflicting Goals) option ‘normal’. In each graph, the number of times the agent chose each of the three top level activities is shown on the y-axis. Each line represents a different activity.
Figure A.4: Case 4 behaviour based on the individual: Action choices for each agent for a particular run of Case 4 (Complex Soft Goal Personality) option ‘normal’. In each graph, the number of times the agent chose each of the three top level activities is shown on the y-axis. Each line represents a different activity.
A. BEHAVIOUR OVER TIME

Figure A.5: Case 5 behaviour based on the individual: Action choices for each agent for a particular run of Case 5 (Different Soft Goal Personalities) option ‘normal’. In each graph, the number of times the agent chose each of the three top level activities is shown on the y-axis. Each line represents a different activity.
Appendix B

Learning A Functional Soft Goal

In this appendix we have further sample graphs for research sub-question 1b: Can characters learn about specific, functional goals? The graphs show how characters learn about two specific goals: “be close to friends” and “don’t be close to enemies”. Characters should be learning to move towards ‘friends’ and away from ‘enemies’.

In Figure B.1, for example, we show the number of times an example character, Anna, choose each of the six plans:

- Move towards friend;
- Move away from friend;
- Move towards neutral;
- Move away from neutral;
- Move towards enemy;
- Move away from enemy.

In the figures each line represents the number of times that Bec chose that particular plan: “move towards” or “move away”. There is a figure for each of the categories of ‘friend’, ‘neutral’, or ‘enemy’.
B. LEARNING A FUNCTIONAL SOFT GOAL

![Graphs showing action choices for Anna and Bec in different scenarios](image)

**Figure B.1:** Action choices for Anna for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.

**Figure B.2:** Action choices for Bec for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.
Figure B.3: Action choices for Chloe for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.

(a) Move with respect to a ‘friend’  
(b) Move with respect to a ‘neutral’ person  
(c) Move with respect to an ‘enemy’

Figure B.4: Action choices for Deb for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.

(a) Move with respect to a ‘friend’  
(b) Move with respect to a ‘neutral’ person  
(c) Move with respect to an ‘enemy’
B. LEARNING A FUNCTIONAL SOFT GOAL

Figure B.5: Action choices for Elle for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.

Figure B.6: Action choices for Fran for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.
Figure B.7: Action choices for Gina for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.

Figure B.8: Action choices for Heidi for a particular run of Case 4 (Complex Soft Goal Personality) mode ‘normal’.
B. LEARNING A FUNCTIONAL SOFT GOAL
Appendix C

Domain-dependent Results

In the domain used for implementation, characters used two main opinions as part of their domain-dependent knowledge, attraction towards others and happiness. Based on attraction towards others, characters were able to build friendships that were relatively stable. Although how friendships are developed is based on domain-dependent equations, whether they wanted to make friends was part of their soft goals and therefore somewhat dependent on the stability of the model. The characters also had a happiness value that represented how close they were to achieving all of their soft goals.

In this appendix, we begin by examining happiness and then friendships as generated in the ‘normal’ mode when characters were able to adapt and use contexts. We finish by discussing the implications of these domain-dependent results.

C.1 Happiness

Happiness is closely related to reward values, since happiness is updated every time the character calculates a personal reward value (see evaluation process Section 3.2.2, page 90). By looking at the happiness graphs, we can also consider how personal reward fluctuated over time during the simulation. We consider two representative Cases as shown in Figure C.1 (page 206). These graphs show happiness intensity as a dot at each time tick. The reference reward is shown as the line, where reference reward is a reflection of all past rewards. Happiness fluctuates from high to low levels often and very quickly over time. The reference reward is also fluctuating, but not as severely.
C. DOMAIN-DEPENDENT RESULTS

In Case 1 (Figure C.1(a)), the average reference reward is higher than Case 4 (Figure C.1(b)). This was discussed in the results presented in Section 5.2 (page 160).

These results indicate that the personal reward and happiness fluctuate a lot, which is perhaps why the characters found it so difficult to learn. Even when the simulation was run for longer periods of time, the reward does not stabilise. We tried changing parameters within the learning function to improve stability but this seemed to have little effect. At the start of the simulation, the characters seemed to “learn” quickly since time steps allowed the character to complete each activity multiple times. These results show that the learning function needs significant work if it is considered desirable for reward and happiness to be more stable. However, how happy someone is can change throughout the day and so perhaps an unstable happiness value could be more realistic than a stable fixed happiness, depending on the domain and the intended use.

C.2 Friendships

In this section, we examine the results based on opinions the character stored relating to each other character, attraction towards others. In our domain, friendships were formed based on insults given and received. The equations that generate attraction towards others are given earlier (see Section 4.1.2.2, page 111). The key ways that characters changed opinions of others was based on not liking people who insult you, liking people who agree with you, liking people talking to you, and liking people who insult people
you do not like. In an effort to simplify the complexity of these relationships, all characters used the same methods to determine how to update opinions.

Attraction towards others allowed characters to classify the other characters as ‘friends’, ‘enemies’ or ‘neutral’. These values were used when the characters decided who they wanted to move towards (or away from), who they wanted to talk to and who they wanted to insult. We will look at five sample runs for each of the five Cases for the ‘normal’ mode (i.e. using our full model with adaptation and contexts). By examining attraction towards others, we can consider how stable the friendships were between characters. We examine graphs of the attraction values held by each individual character towards the others, and also the friendship networks. We begin by explaining how to read the figures.

In this section, we show a graph for each of the eight characters of attraction towards others for each of the five Cases, taken from a single, sample run. For example, Bec’s attraction towards others over time is shown in Figure C.2(b). Each of the seven lines represents a different character, for example, Heidi. The attraction towards that character is on a scale of $[-1, +1]$, with $-1$ being the worst value, and $+1$ the highest. If the attraction towards others goes over the threshold of $+0.3$ (as set by the domain-dependent threshold from emotionality, Section 4.2.2 page 122), then Heidi will be considered a ‘friend’ by Bec. If the attraction value goes below -0.3, then Heidi will be considered an ‘enemy’.

We also show a friendship network representing a snapshot in time of how the characters feel towards each other. In particular, we show the friendship network as it stands at the end of the simulation. In these diagrams, (such as in Figure C.2(k)) an arrow from one character to another indicates that the originating character considers the other to be a ‘friend’. 
C. DOMAIN-DEPENDENT RESULTS

Figure C.2: Friendship network and attraction levels for Case 1 ‘normal’.
C.2 Friendships

C.2.1 Case 1: Clear Preference Against One Activity

We start by considering the Case where there was a clear preference against one activity, insults. Our domain was set up so that it would be difficult to make friends directly. Therefore, in our domain, generating or listening to insults were the only ways that characters could make friends, since these plans are the only ones that update the attraction towards others value (see Section 4.1.2.2, page 111). In Figure C.2, we see the graphs from one example run (the friendship networks generated were fairly similar in all ten runs). The characters did not make very many ‘friends’ and, in almost all runs, there were no mutual friendships generated (where both characters consider the other to be a ‘friend’). For example, the network in this single run (Figure C.2(k)), shows arrows (i.e. friendships) in only one direction. Only Anna and Bec consider someone to be their friend. The low number of friendships in Case 1 is likely due to the clear preference against insults, and consequently against making friends.
C. DOMAIN-DEPENDENT RESULTS

Figure C.3: Friendship network and attraction levels for Case 2 ‘normal’.
C.2.2 Case 2: Multiple Ways to Achieve Goals

The sample graphs for Case 2 are shown in Figure C.3. Here, the friendship network generated at the end of the simulation (Figure C.3(k)) is more complex than in Case 1. Three mutual friendships are formed and overall more characters have “friends”. For example, as seen in Figure C.3(b), Bec considers Fran to be a ‘friend’ beginning near the start of the simulation and does not change her opinions very much. In Figure C.3(f) Fran also becomes friends with Bec early in the simulation and this opinion remains fairly stable. However, Fran also changes her attraction to Chloe and Elle as the simulation continues, so that she becomes friends with both of them by the end.
C. DOMAIN-DEPENDENT RESULTS

Figure C.4: Friendship network and attraction levels for Case 3 ‘normal’.  

212
C.2 Friendships

C.2.3 Case 3: Conflicting Goals

The attraction values and friendship networks for Case 3 are shown in Figure C.4. The final network shows four characters (Deb, Elle, Gina and Heidi) as a core group who mostly like each other and who are liked by other characters. Bec, Chloe and Fran are outsiders who like other individual characters, but whom no one likes. Anna considers no one her friend, neither does anyone consider her their friend. When we examine what activities the characters preferred to do (Figure 5.3, page 149), there does not appear to be a correlation meaning that, characters who always prefer to insult others are sometimes popular, like Gina, and sometimes unpopular, like Anna. If we examine the actual fluctuations of their attraction levels over time (for example, Figures C.4(e) and C.4(h)), we see that the levels are fairly constant. Once one character “likes” another character, they continue to do so and do not change their mind often. However, this is not as true for Elle and Fran who change their opinions slightly throughout the scenario.
C. DOMAIN-DEPENDENT RESULTS

Figure C.5: Friendship network and attraction levels for Case 4 ‘normal’.
C.2.4 Case 4: Complex Soft Goal Personality

The graphs from Case 4, where characters are pursuing many soft goals, are shown in Figure C.5. The characters are trying to achieve “make friends” and “don’t make enemies” as well as four other soft goals. The friendship network (as shown in Figure C.5(k)) and the individual character graphs show that, although some friendships were formed, the characters were not very successful at their goals. That is, in most Cases, the characters had more ‘enemies’ than friends. It is interesting that, according to the friendship network, two groups of friends were formed. Also, Fran appears to be the most popular girl, since three other characters like her, and yet she only likes Anna. These apparent abnormalities happen in the real world as well.
C. DOMAIN-DEPENDENT RESULTS

Figure C.6: Friendship network and attraction levels for Case 5 ‘normal’.
C.2.5 Case 5: Different Soft Goal Personalities

In Case 5, characters with different soft goal personalities, a relatively complex friendship network was generated (see Figure C.6(k)). In this network, we see that Gina and Heidi are the two most popular girls. Interestingly, the only soft goal that both Gina and Heidi were attempting to achieve was “make friends”. So, according to the friendship network both girls achieved their goal quite well, even though Heidi considered only Gina to be her friend. The friendship network shows that there are some characters, such as Elle and Deb, who like three other characters but are liked by no one. This can happen in the real world, particularly with girls of school age.
C. DOMAIN-DEPENDENT RESULTS


REFERENCES


222


REFERENCES


REFERENCES


REFERENCES


REFERENCES


228


REFERENCES

*Meets the Machine*, 173–202, Oxford University Press, New York, USA. Referred to on p. 31, 32


REFERENCES


Sandercock, J., Papasimeon, M. & Heinze, C. (2004). An agent, a Bot and a CGF walk into a bar... In *Proceedings of SimTecT 2004*, [May], Canberra, Australia. Referred to on p. 29

REFERENCES

Agents, 6th International Conference, IVA 2006, [August, Marina Del Rey, California, USA], vol. 4133 of Lecture Notes in AI, 357–368, Springer, Berlin, Germany. Referred to on p. 103


REFERENCES


SILVERMAN, B.G. & BHARATHY, G.K. (2005). Modeling the personality and cognition of leaders. In Behavior Representation in Modeling and Simulation (BRIMS), [May, Universal City, California, USA], SISO (IEEE), Orlando, Florida, USA. Referred to on p. 53, 55, 57


REFERENCES


ence on Autonomous Agents, [June, Barcelona, Spain], 365–372, ACM Press, New York, USA. Referred to on p. 38, 55, 56, 61, 62, 63, 64.