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[1]. 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

[1]Examples 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.

4.1 Implementation

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.

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

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 difficultly 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 unhappy, 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 happiness by incorporating soft goals to define their goal state.

Results from the pilot implementation indicated that the characters had some diversity, 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 identified 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

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

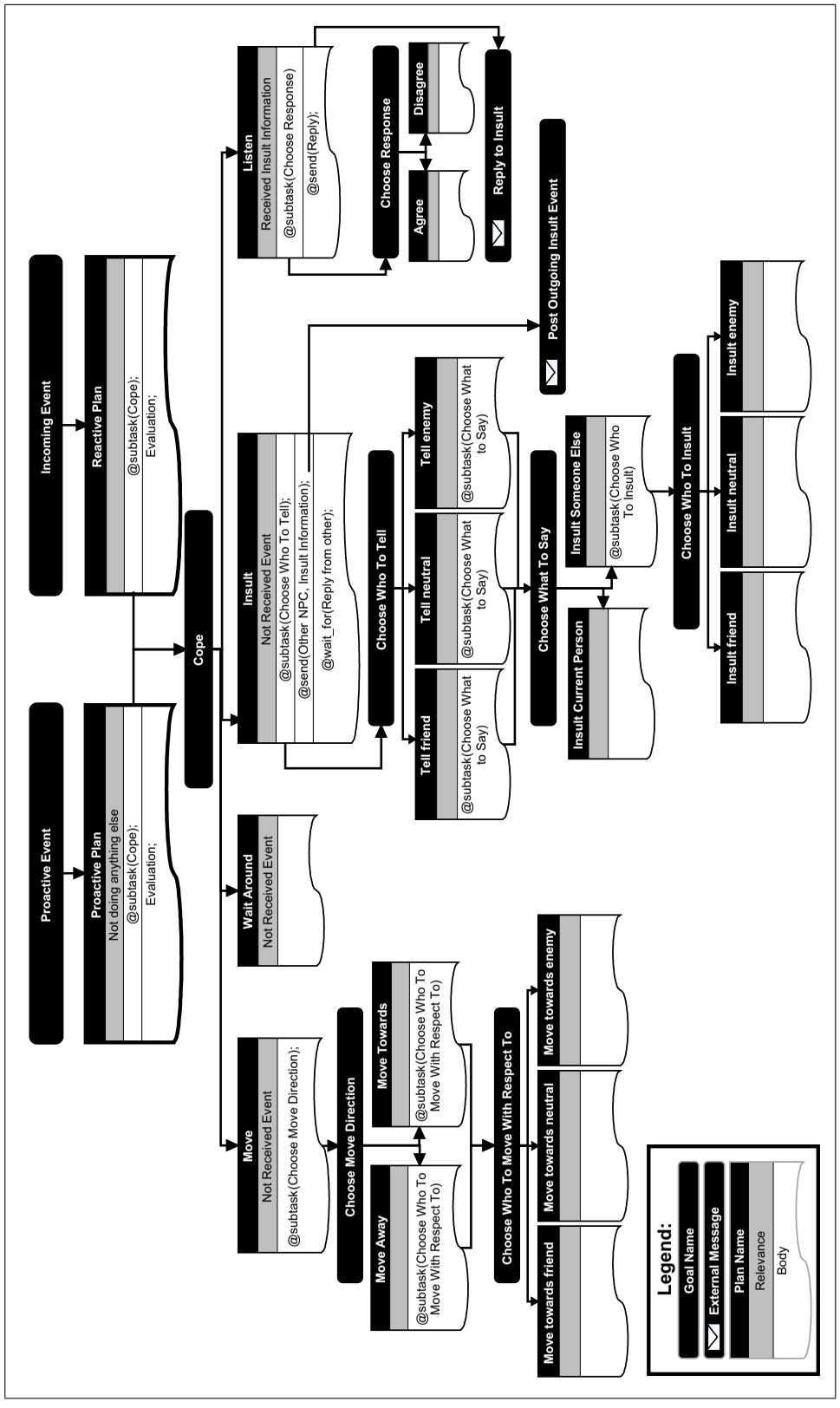

Figure 4.2: Implemented domain-dependent goal/plan hierarchy used for all characters.

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

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-

tially 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".

- 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[1]. 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 ($\frac{1}{8}$th) 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

[1]The 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.

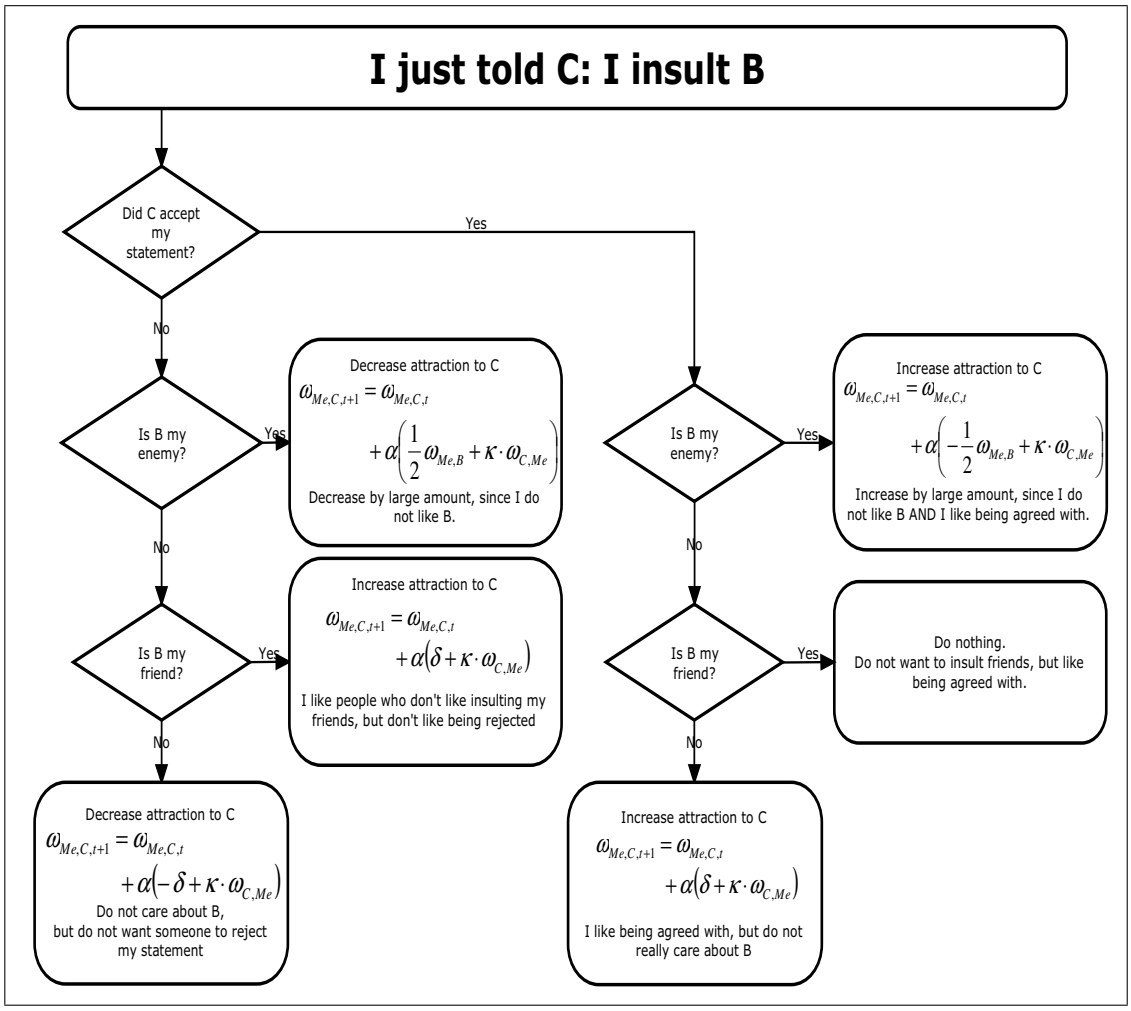

Figure 4.6: Flowchart used to determine how to update attraction values when the character ("Me") has *insulted someone*. Terms used:

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

$\omega_{C,Me}$ 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");

α is the same step-wise learning parameter as used to update the reference reward;

δ is a small value; and

κ is a constant weight on the importance the character gives to the other character's attraction value.

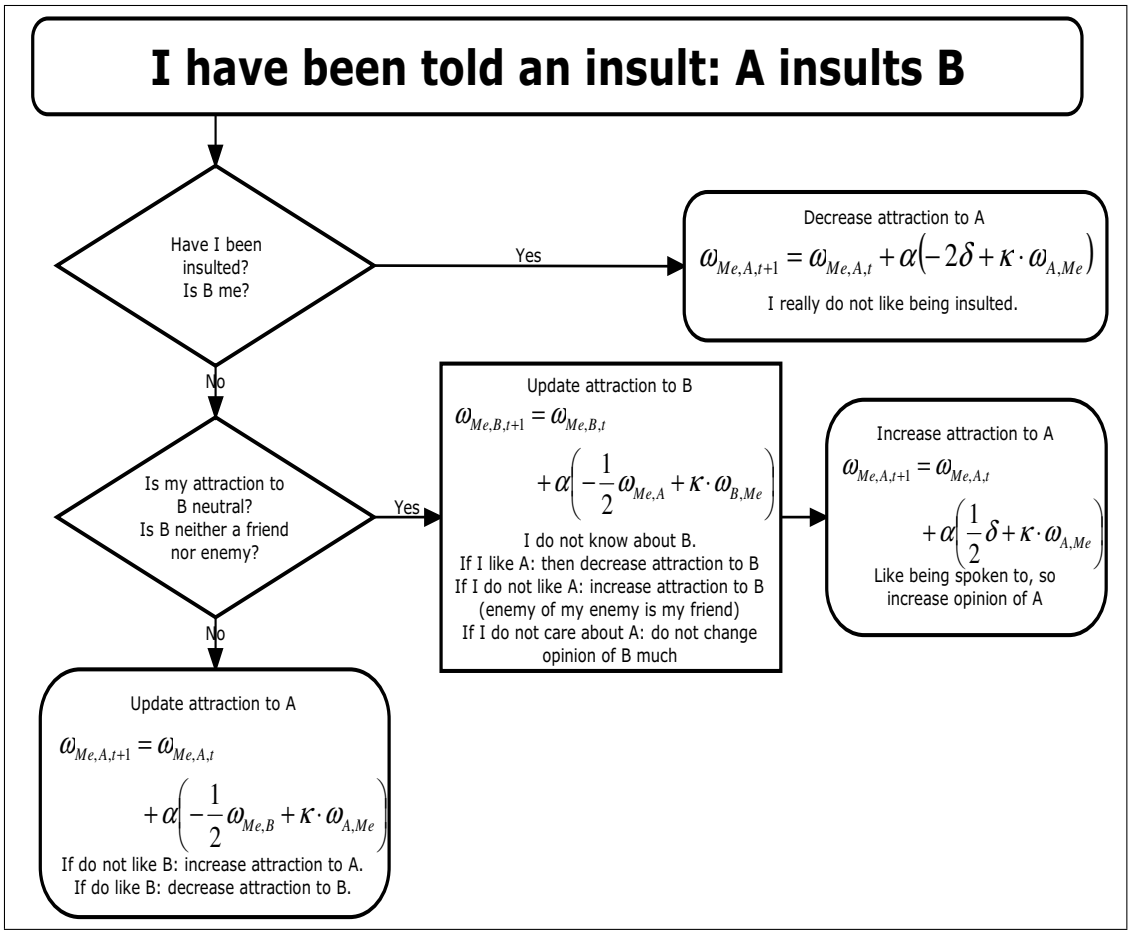

Figure 4.7: Flowchart used to determine how to update attraction values when the character ("Me") has *been told an insult*. Terms used:

$\omega_{A,B,t}$ 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");

α is the same step-wise learning parameter as used to update the reference reward;

δ is a small value; and

κ 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

		Character A		
		Like (+1)	Don't Care (0)	Don't Like (-1)
Character B	Like (+1)	2	1	0
	Don't Care (0)	1	0	-1
	Don't Like (-1)	0	-1	-2

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_{total}. The maximum possible number of friend points the character can have is $n_{max} = 2 \times n_{characters}$, where $n_{characters}$ 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:

$$z_{friend} = \frac{n_{total}}{n_{max}} \tag{4.1}$$

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_{mutual}.
  2. Insults that this character said, but the other character did not agree with, n_{Isaid}.
  3. Insults that the character was told, but did not agree with, $n_{theySaid}$.

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:

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

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

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

$$z = \frac{n_{insulted}}{N} \tag{4.3}$$

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

$$z = \frac{n_{close}}{n_{total}} \tag{4.4}$$

  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_{met} [1] and the total number of other characters in the simulation, n_{total}:

$$z = \frac{n_{met}}{n_{total}} \tag{4.5}$$

  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

[1] If an attraction value for a character has not been initialised, then the character has not been met yet.

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[1]. 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]See JACK documentation for details. Accessible through `http://www.agent-software.com.au/products/jack/documentation_and_instructi/index.html`

```
Time,EvalNum,Reward,RefReward
6057,646,0.3214285714285714,-0.2995583838991689
6066,648,-0.34523809523809523,-0.20641034060000785
6067,647,0.3214285714285714,-0.22723450379572097
6083,649,-0.34523809523809523,-0.14493504251207712
6093,650,-0.34523809523809523,-0.17498050042097985
6103,651,-0.34523809523809523,-0.20051913964354715
6109,653,-0.6785714285714286,-0.22222698298272936
6114,652,-0.34523809523809523,-0.29067864982103425
6127,655,-0.6785714285714286,-0.2988625666335934
6139,654,-0.6785714285714286,-0.3558188959242687
6150,657,-0.6785714285714286,-0.4042317758213427
6150,656,-0.6785714285714286,-0.44538272373385557
6170,658,-0.34523809523809523,-0.48036102945949155
```

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

```
<SomaticMarkers__Tuple
 :context  "HL"
 :planName  "Move"
 :preferenceValue  0.258688
 :updatedCount  17
>
<SomaticMarkers__Tuple
 :context  "HM"
 :planName  "Move"
 :preferenceValue  2.150914
 :updatedCount  91
>
<SomaticMarkers__Tuple
 :context  "HM"
 :planName  "Insult"
 :preferenceValue  0.785287
 :updatedCount  44
>
```

```
<SomaticMarkers__Tuple
 :context  "HM"
 :planName  "WaitAround"
 :preferenceValue  5.352875
 :updatedCount  224
>
<SomaticMarkers__Tuple
 :context  "LM&Towards"
 :planName  "MoveWRTEnemy"
 :preferenceValue  -0.380345
 :updatedCount  12
>
<SomaticMarkers__Tuple
 :context  "LM&Neither"
 :planName  "DisagreeToInsult"
 :preferenceValue  -0.454140
 :updatedCount  25
>
```

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:

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

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,

$$d = \frac{c}{c_{\mathrm{TOT}} \cdot b} \tag{4.7}$$

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_{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:

$$p_{t+1}(a_t) = p_t(a_t) + \beta(r_t - \bar{r}_t) \tag{4.8}$$

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".

For example, if the range of the value is $(0, 1)$, then the label is "H" when it is above $0.5 + 0.3 * 0.5 = 0.65$. The label is "L" when the value is less than $0.5 - 0.3 * 0.5 = 0.35$ 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 σ 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, ϵ 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 $\epsilon \leq \frac{1}{4}b$ (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

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

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

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.

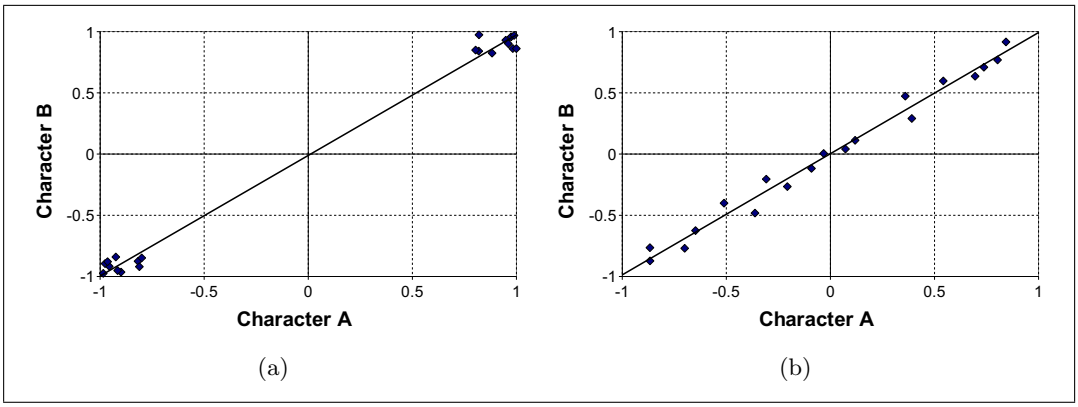

Figure 4.10: Two examples of how characters can be the *same* for one activity: based on frequency of choice of the activity over regular time intervals. Line represents best fit line.

This makes clustering prone to human error. Given all of these factors, clustering was excluded as a possible measure of individuality [1].

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.

[1] These conclusions are based on research carried out specifically for this thesis by Andrew Buelke and Kaye Marion of the RMIT Statistics Department.

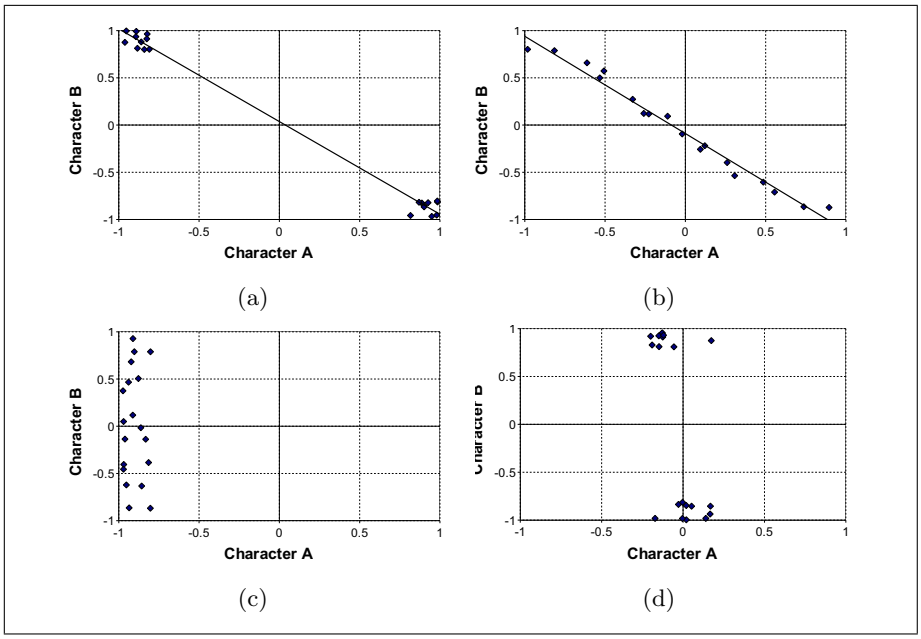

Figure 4.11: Four examples of how characters can be *different* for one activity: based on frequency of choice of the activity over regular time intervals. Line represents best fit line.

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

4.11(c), 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:

Name	Move	Wait	Insult
Anna	5	5	5
Bec	10	10	10
Chloe	5	10	5

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.

Name	Move	Wait	Insult
Anna	33.3%	33.3%	33.3%
Bec	33.3%	33.3%	33.3%
Chloe	25%	50%	25%

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:

$$C_2^n = \frac{n!}{2!(n-2)!} \tag{4.9}$$

That is, the number of possible differences between characters for a particular activity is C_2^n. 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 $2 \times C_2^n$. 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.

		Adaptation	
		On	Off
Context	On	3. Normal	N/A
	Off	2. Context Off	1. Adaptation Off

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.

4.2.4.2 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.
4. Complex Soft Goal Personality.
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

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

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.

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.

Research Sub-question	Case				
	1	2	3	4	5
1a: Behaviour Over Time	Success	Success	Success	Success	Success
1b: Learning A Functional Soft Goal	N/A	N/A	N/A	Success	N/A
2a: Behaviour in Different Contexts	Likely Fail	Likely Success	Success	Likely Success	Likely Success
3a: Chi-squared Test	Likely Success	Success	Success	Success	Success

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. 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

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, 38) 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.

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

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. 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.

1. Clear Preference Against One Activity: Likely Fail.
2. Multiple Ways to Achieve Goals: Likely Success.
3. Conflicting Goals: Likely Success.
4. Complex Soft Goal Personality: 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.