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. 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.

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

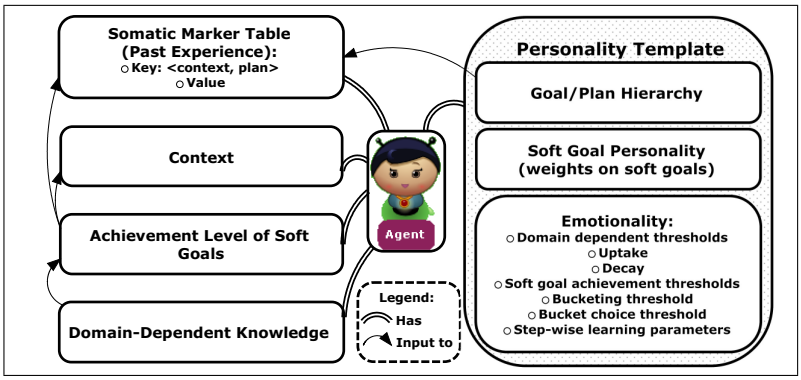

Figure 3.1: Agent components and beliefs

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

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

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

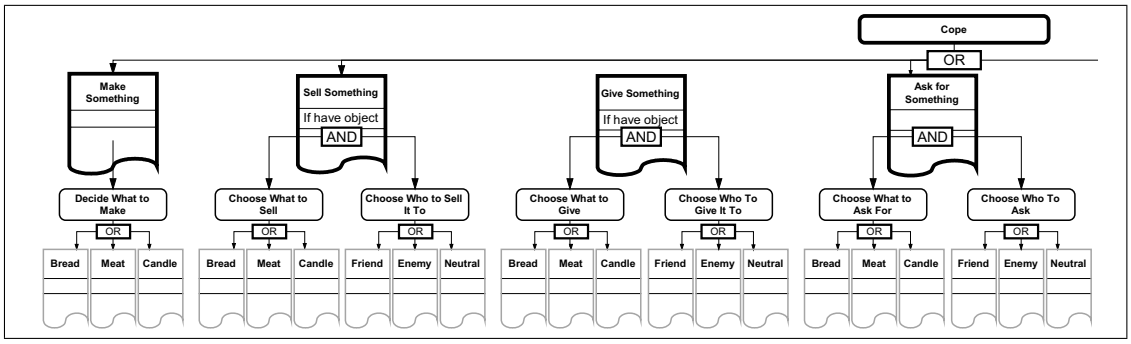

Figure 3.2: Possible partial goal/plan hierarchy from the village domain. Part A.

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

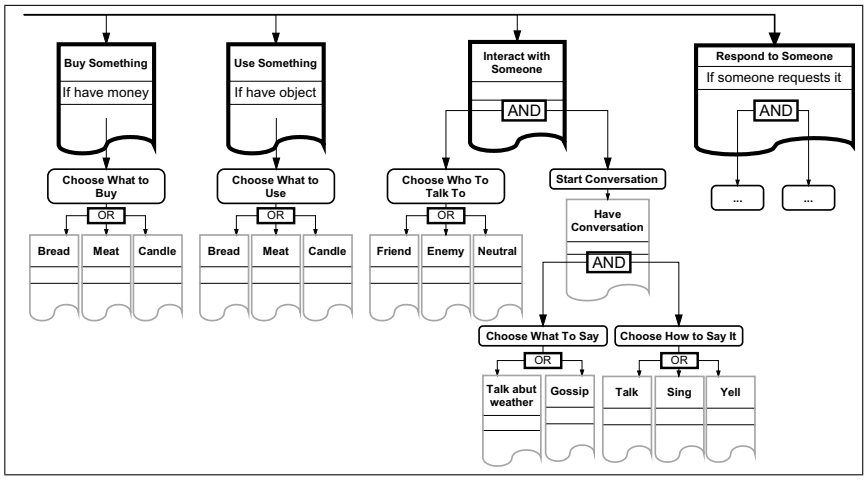

Figure 3.3: Possible partial goal/plan hierarchy from the village domain. Part B.

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.

3.1.3.3 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;

3.1.4 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.

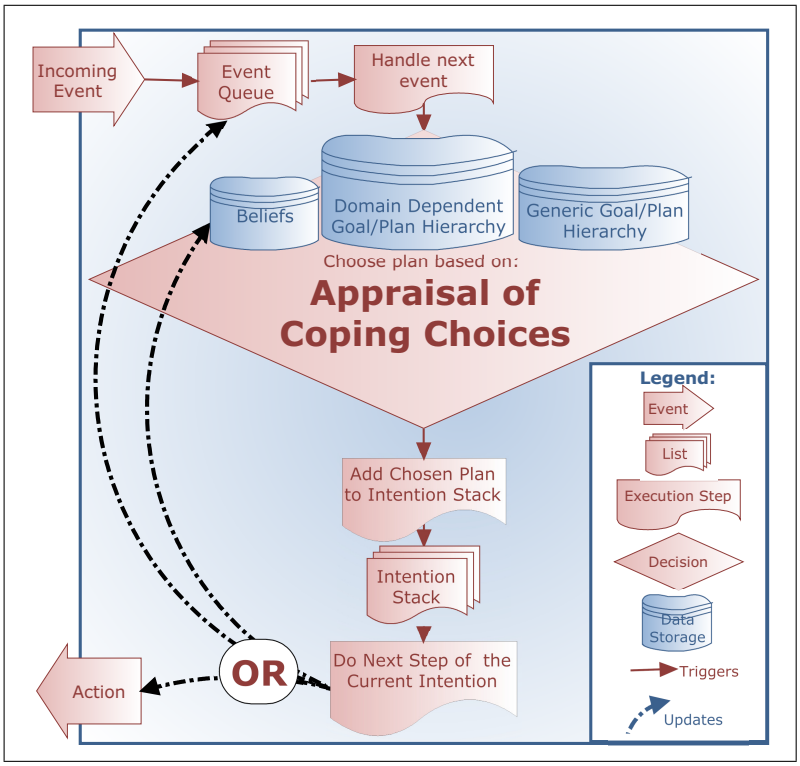

Figure 3.4: Expanded version of the BDI execution loop: The standard BDI execution loop expanded to use appraisal of coping choices based on somatic markers.

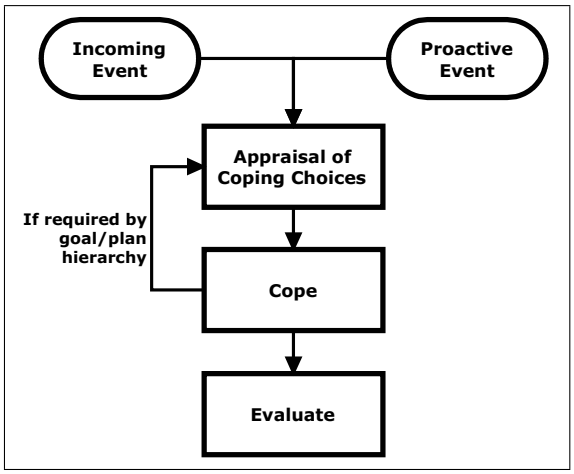

Figure 3.5: Adaptation process: triggered by an incoming or self-generated event.

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.

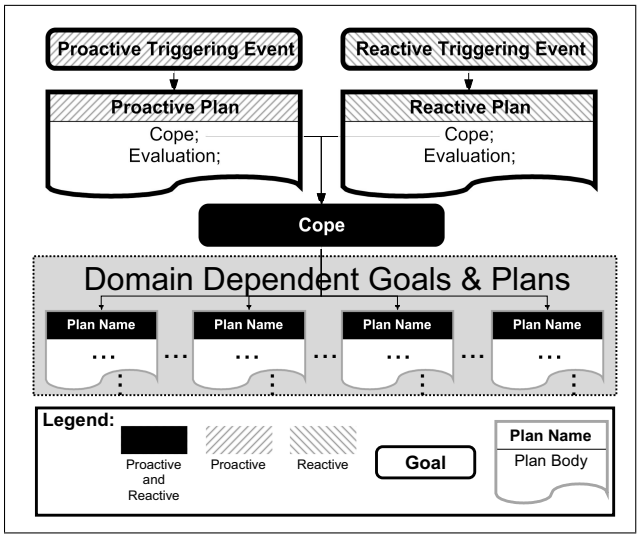

Figure 3.6: The adaptation process with a generic goal/plan hierarchy: linking the adaptation loop to the top-level of a domain-dependent BDI hierarchy.

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.

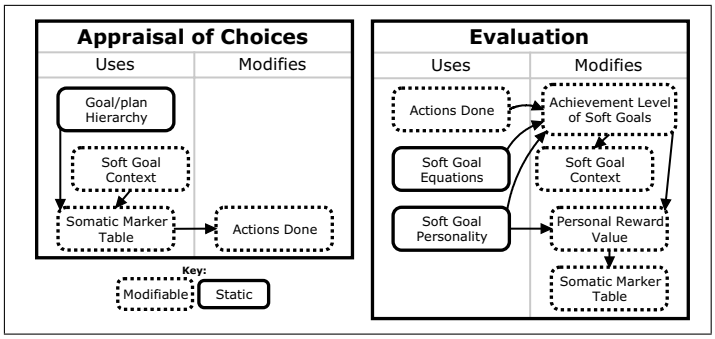

Figure 3.7: Effect of adaptation processes on components and beliefs: How components and beliefs affect and are affected by the two main adaptation process, i.e. appraisal of choices and evaluation.

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 sub-processes 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 36). 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

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, σ, for applicable plans. We also use the agent's bucketing thresholds, δ_{-ve} and δ_{+ve}, from emotionality in the personality template. The plans are grouped into "desirable", "undesirable", and "don't care" based on:

- if $p_{s,t}(a_i) > \bar{p} + \sigma\delta_{+\text{ve}}$; plan is in the "desirable" group;
- if $p_{s,t}(a_i) < \bar{p} - \sigma\delta_{-\text{ve}}$; plan is in the "undesirable" group; and
- if $\bar{p} - \sigma\delta_{-\text{ve}} \leq p_{s,t}(a_i) \leq \bar{p} + \sigma\delta_{+\text{ve}}$; 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, ϵ, 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, τ, guides the use of ϵ according to:

   - if $\epsilon \geq \tau$, try to choose in the order:
     1. "desirable"; 2. "neither"; 3. "undesirable";
   - if $\epsilon \leq \frac{1}{4}\tau$, (i.e. a small number and very unlikely to occur), try to choose in the order:
     1. "undesirable"; 2. "neither"; 3. "desirable";
   - else, try to choose in the order:
     1. "neither"; 2. "desirable"; 3. "undesirable";

   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.

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.

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.

A domain-dependent reward function, based on the distance to the ideal soft goal value, $Z_{j,\max}$, 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, $r_{\mathrm{ABS}t,j}$, at time t (i.e. now) for each soft goal, j, is:

$$r_{\mathrm{ABS}t,j} = Z_{j,\max} - z_{t,j} \tag{3.1}$$

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

$$r_{\mathrm{ABS}t,j} = z_{t,j} - Z_{j,\max} \tag{3.2}$$

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 $r_{min} \leq r_{t,j} \leq r_{max}$ (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):

$$r_{t,j} = r_{max} - (r_{max} - r_{min}) \times \frac{Z_{j,max} - z_{t,j}}{Z_{j,max} - Z_{j,min}} \tag{3.3}$$

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:

$$r_t = \frac{\sum r_{t,j} \cdot w_j}{\sum w_j} \tag{3.4}$$

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:

$$p_{s,t+1}(a_i) = p_{s,t}(a_i) + \beta(r_t - \bar{r}_t) \tag{3.5}$$

where β 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:

$$\bar{r}_{t+1} = \bar{r}_t + \alpha(r_t - \bar{r}_t) \tag{3.6}$$

where α $(0 < \alpha \leq 1)$ 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:

$$p_{s,t+1}(a_i) = p_{s,t}(a_i) + \beta \cdot d \cdot (r_t - \bar{r}_t) \tag{3.7}$$

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

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

$$d = \frac{c}{c_{\text{TOT}} \cdot b} \tag{3.8}$$

- β 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:

$$I(k, t_0) = r_t \times u \tag{3.9}$$

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.

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:

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

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:

$$I(t) = \sum_k I(k,t) \tag{3.11}$$

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:

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

where

- α 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

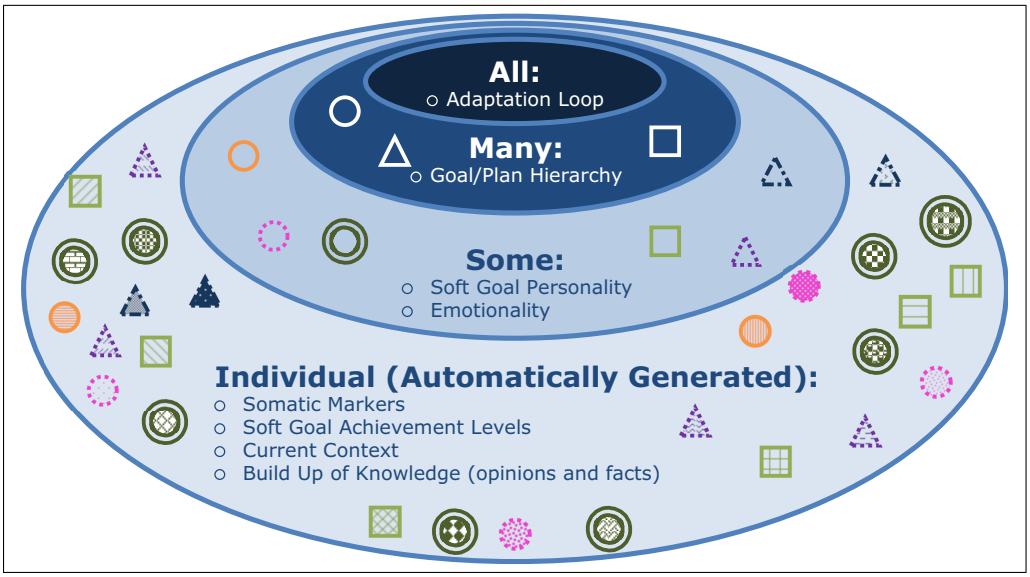

Figure 3.8: What makes an individual an individual in our model: Components that only one agent has compared to components and processes that all agents have.

(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.