Folk Psychology for Human Modelling: Extending the BDI Paradigm

Article · July 2004

DOI: 10.1109/AAMAS.2004.129 · Source: CiteSeer

1 author:

Emma Norling
Manchester Metropolitan University

Folk Psychology for Human Modelling: Extending the BDI Paradigm

Emma Norling
Department of Computer Science
and Software Engineering
The University of Melbourne
norling@acm.org

Abstract

BDI agents have been used with considerable success to model humans and create human-like characters in simulated environments. A key reason for this success is that the BDI paradigm is based in folk psychology, which means that the core concepts of the agent framework map easily to the language people use to describe their reasoning and actions in everyday conversation. However there are many generic aspects of human behaviour and reasoning that are not captured in the framework. While it is possible for the builder of a specific model or character to add these things to their model on a case by case basis, if many models are to be built it is highly desirable to integrate such generic aspects into the framework. This paper describes an approach to extending the BDI framework to create an enhanced framework for human modelling. It draws upon the folk psychological roots of the framework to create the extension, maintaining the mapping between the knowledge representation in the framework and the natural means of expressing expert knowledge. The application of this approach is illustrated with an extension to support human decision making.

1. Introduction

The benefits of using a BDI(belief-desire-intention)-based agent programming language for human modelling were first noted early in the application of this technology to military simulation. Researchers noted that the close match between the core BDI concepts and the natural terminology that the experts being modelled used to describe their reasoning facilitated knowledge capture, representation and debugging [9]. At the same time, it was being noted that the BDI framework provided a very high-level abstraction of human reasoning, and that there were many aspects of human reasoning and behaviour that were *not* captured by the framework, which needed to be explicitly coded by the model builders. It would be possible to extend the framework to address these aspects of human reasoning, but the risk was that in doing so, the benefits given by the usability of the framework would disappear. By taking advantage of the folk psychological roots of BDI, it is possible to maintain this benefit of the the framework while capturing more of the domain-independent aspects of human reasoning and behaviour, making the framework more suited to human modelling.

The notion of capturing human reasoning using beliefs, desires and intentions, as the BDI agent framework does, is an extremely high level abstraction of human reasoning. This abstraction ignores many of the details of human reasoning that influence the choices that are made and actions that are taken. However this does not mean that it is not possible to explain these details using folk psychology, indeed many different aspects of human reasoning and behaviour have folk psychological explanations. If such explanations can be integrated into the existing framework, the resulting framework becomes better suited to human modelling, because it captures more of the generic details of human behaviour.

The approach presented in this paper is to incrementally enhance the framework, taking a folk psychological explanation of a characteristic of interest and integrating this into the framework. Such explanations generally refer to the same or similar concepts as are already used in the framework, but add a level of complexity, including, sometimes, additional concepts. These concepts though should still be ones that people naturally use to explain their reasoning or behaviour, since they come from a folk psychological explanation. However sometimes the implementation of these explanations requires the introduction of 'artificial' constructs, when what seems like a relatively simple explanation is not so simple to implement. In such cases, the value of the extension must be carefully considered: does the additional modelling power introduced by the addition of the characteristic justify the shift away from a 'natural' form of representation? The implemented example described in section 4 is one case that does not seem to suffer from the addition of some of these 'artificial' parameters, since the agent's behaviour is relatively insensitive to their settings, so the model builders can for the most part ignore them, using the default values.

The remainder of this paper is structured as follows: First the BDI framework is discussed, with particular emphasis on its use in human modelling and the creation of human-like synthetic characters. Section 3 then discusses the folk psychological approach to extending the framework, which is illustrated with a specific example in Section 4. Finally, Section 5 presents the limitations of this approach, with some pointers to how they can be overcome.

2. BDI Agents and Human-Like Synthetic Characters

The BDI framework is based upon a folk-psychological view of reasoning, that is, the way people *think* that they think, as opposed to the actual mechanism of the brain (if indeed such things are understood). While folk psychology suffers criticism because of the necessity for introspection (as opposed to an objective understanding), it does provide a robust mechanism for reasoning about human reasoning. Bratman's *Intention, Plans and Practical Reason* [2] best summarises the philosophy of the BDI framework (and indeed has been the basis for much work in this field), but an abbreviated summary is given here.

Briefly, an agent is characterised by its beliefs, goals (desires), and intentions — it will *intend to do* what it believes will *achieve its goals* given its *beliefs about the world*. As well as these three components, a BDI agent is usually assumed to have a *plan library* — a set of 'plans as recipes' that it can use to achieve particular goals given particular preconditions. An intention is formed when the agent commits to a particular plan — a particular sequence of steps to perform — from this set in order to achieve a goal. The steps themselves may be atomic actions, or they may be subgoals, which can be satisfied by other plans. The beliefs, goals and intentions of an agent are maintained by the BDI reasoning engine. This engine is what drives the agent, updating beliefs, monitoring and updating goals and intentions, selecting plans to achieve goals, and based on the current intentions, selecting the actions to perform.

A key feature of the plan library is that although the plans are fixed 'recipes' for action, they do not have to be fully specified. For any particular goal, there may be multiple plans to achieve that goal, and while any plan *may* be fully specified as a sequence of actions, a plan may instead consist of a sequence of subgoals, or a combination of actions and subgoals. In the case that the plan contains subgoals, the agent can delay the choice of how to achieve a particular subgoal until the time that it reaches that stage of the plan. While this does not achieve the full range of adaptability that people display, it does allow considerable flexibility in the agent's planning, and its resulting behaviour, and also allows a balance between reactive and deliberative planning.

2.1. BDI-Based Synthetic Characters

BDI agents have been used to model human behaviour and create human-like characters in a range of applications. Perhaps the largest use of BDI agents for this purpose has been in the area of military simulation, where BDI agents have been used to model the personnel in synthetic environments. Examples of this type include agents to model fighter pilots [9], Airborne Early Warning and Control (AWAC) crew [4] and military commanders [7]. Other applications include the simulation of users for interface evaluation [13], and the creation of the key character in the popular computer game *Black&White* [11].

It is precisely because this philosophy has its foundations in folk psychology that it is useful in capturing human knowledge. When asked about how they think about a problem, people have a tendency to explain their actions in terms of their intentions, which in turn are explained in terms of their goals and beliefs. Moreover, when they describe the ways in which they try to achieve goals (the plans they use), they do this in a hierarchical manner, which maps to the partial plans needed for the plan library. Extracting this information from people does require careful planning and structured questioning, but the fact that model builder and the subject being modelled are referring to the same concepts does simplify matters.

2.2. Why Extend the BDI Framework?

Although the BDI agent framework is well suited to human modelling, it could be more so. There are many generic aspects of human behaviour that are not incorporated into the framework, and existing applications that use BDI agents for human modelling either ignore these aspects of behaviour or program them explicitly into each model/character that is built. *Black&White's* creature, for example, is BDI-based, but also incorporates other human-like characteristics such as hunger, desire for entertainment, and a reinforcement learning mechanism which allows the user to 'teach' it [11].

One of the largest communities of developers of BDI agents for military simulation is situated within Australia's Defence Science and Technology Organisation (DSTO). This group was one of the early adopters of the technology [12], and they have an ongoing interest in developing BDI-based models of human behaviour. In 2000, these developers were surveyed as to what were the shortcomings of

BDI agents for the types of human modelling applications that they developed. They identified a range of human characteristics that could be incorporated into the framework. If the framework were extended to model these characteristics, it would either 1) mean that developers didn't have to explicitly encode them into every model, or 2) mean that the constraints on the applications could be expanded. The characteristics that were identified included:

• decision making	• timing
• expertise	• inaccuracies
• workload effects	• memory
• emotion	• situation awareness
• fatigue	• learning

It should be noted that although the developers were asked to consider what was important in *their* domain of expertise (that is, military simulation), most (if not all) of these characteristics are more broadly applicable.

3. A Folk Psychological Approach to Extending the BDI Framework

Folk psychology is 'layman's psychology' — a means of explaining the behaviour of others via their reasoning. Although it makes no attempt to model the low-level details of human reasoning, it is a robust tool for understanding and predicting the behaviour of others. The folk psychological model used in the BDI paradigm is perhaps the most abstract version of folk psychology, simplifying all reasoning to the agent *intending* to do what it believes will achieve its *desires*, given its *beliefs* about the world. Dennett calls this the *intentional stance* [3], and claims that it is routinely used to explain the behaviour of others. However there are also many other folk psychological models that are used to give far more detailed explanations of behaviour, such as explanations for emotional responses (e.g. [15]), or for human decision making strategies (such as the one discussed in Section 4).

The approach advocated here is to take advantage of these explanations, integrating them into the framework. This should maintain the key benefit of the BDI approach to human modelling — the mapping between the framework's concepts and the natural means of expressing reasoning — but allow the framework to grow to encompass more of the basic aspects of human reasoning and behaviour. It is not possible to capture all of the generic aspects of human behaviour in this way, because not all have folk psychological explanations. Many of them do though, and capturing them in the framework will be a significant benefit for human modelling. Those aspects lacking folk psychological explanations will need another approach, such as that which is suggested in Section 5.

One of the challenges of this approach is to determine the mapping between concepts used in different folk psychological explanations. The terminology used in different explanations varies considerably, but in general the core concepts of a given explanation either (1) map directly to existing concepts in the framework or (2) introduce new features into the framework. In this latter case, the new features add complexity to the framework, but (hopefully) this is offset by the improvement this gives to the representation of human behaviour and/or reasoning in the models.

A second challenge of this approach is to ensure that the *implementation* of the folk psychological explanation does not overly complicate the framework. In some cases the explanation might be conceptually simple, but computationally challenging. The example presented in the next section shows how the implementation can create additional constructs over and above the folk psychological explanation.

4. An Application of this Approach: Human Decision Making

To illustrate this approach to extending the framework, one particular example is presented here, of incorporating a human-like decision making strategy into the framework. In theory, BDI agents used a utility-based decision strategy, where decisions are made to maximise the expected utility of the selected course of action. In the implemented BDI-based agent programming language used for this work (JACK Intelligent Agents [1]), the default behaviour is simply to select the first applicable plan, regardless of expected utility. In real-world situations, while either of these strategies may be used on occasion, it is argued that neither is commonly used, particularly when the person is operating within their area of expertise [6].

The literature reveals not one but many decision making strategies. Many researchers, particularly those who study decision making by people in the field rather than in laboratories, argue that people will use different strategies depending on the situation. The extension described here is a first step towards capturing the full range of human decision making. Rather than trying to capture all possible decision making strategies (let alone the meta-level reasoning of when to use which one), the extension captures one strategy which is particularly applicable when modelling experts operating in domains with these characteristics:

- Ill-structured problems
- Uncertain, dynamic environments
- Shifting, ill-defined or competing goals
- Action-feedback loops
- Time stress
- High stakes
- Multiple players
- Organisational goals and norms

The implemented model, known as the *recognition-primed decision* (RPD) model, has been proposed by

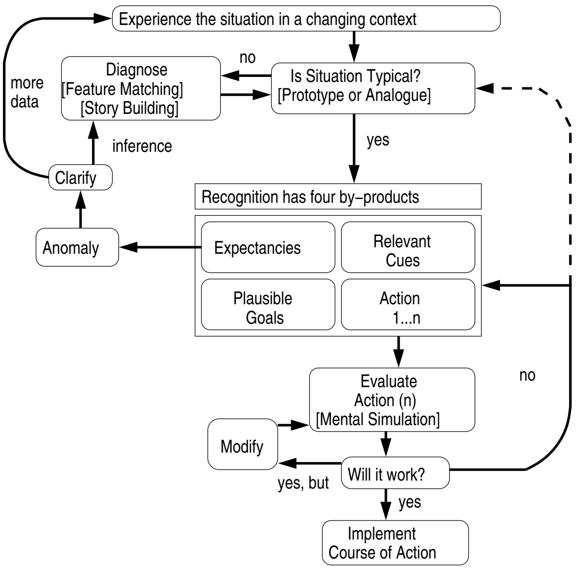

Figure 1. The Recognition-Primed Decision model of decision making

Klein to describe the decision process of experts operating in these types of environments [5]. It has been applied to experts in a range of fields, including fire fighters, intensive care nurses and battle command teams [5]. It is based on the idea that people operating in their field of expertise spend very little time on *decision making*; rather they devote most of their energy to *understanding the situation*, and once the situation is recognised, the choice of course of action is virtually automatic. It assumes that experts learn to recognise subtle differences in situations that recommend one course of action over others. The process is summarised in Figure 1 which shows how a new course of action (plan) is selected. Obviously this diagram shows only a single decision point; the same process is repeated each time a new goal arises.

4.1. Mapping between RPD and BDI

While the developers of the RPD model might cringe at it being called a folk psychological description of decision making, the core concepts in the model are loosely defined terms that are commonly used to describe human reasoning. While they do not correspond exactly to the terms used by the BDI framework, there is a close mapping between the two, as shown in Table 1. The concept of mental simulation has no obvious equivalent in BDI, unless one argues that plans themselves do this. However the key difference between the two models is that in the RPD model, the agent *learns from its mistakes*, learning to recognise the

RPD	BDI
changing context	agents changing beliefs
situation recognition, relevant goals & cues	plan selection
possible actions	applicable set
expectancy violation	plan failure

Table 1. The mapping between RPD and BDI

subtle cues that distinguish one situation from another and hence make one plan preferable over another.

This model of decision making captures some of the adaptability that people exhibit. The model does not allow for the agent to come up with completely new courses of action, however it does allow the agent to vary the way it assembles sub-plans, rather than always picking the same course of action in a given situation. It also helps deal with one of the difficulties involved in knowledge elicitation. In general, people are good at describing the plans that they use, but specifying the conditions under which these plans are useful is a different matter: they often fail to mention subtle but important differences. Because a RPD-enabled agent can learn to recognise different situations, it can compensate for this problem (so long as it is acceptable to have the occasional failure from which it can learn).

4.2. Design of the Extension

To incorporate this model of decision making into the BDI framework, the key modification that needed to be made was the ability to learn to recognise situations and select the appropriate plan based upon this. In JACK, the normal filter for applicable plans is based on the *context conditions* — a boolean test that is defined at the time the plan is written. There is no mechanism in JACK to allow the context condition to be dynamically updated at run time. Instead, the meta-level reasoning capability of JACK was used, which allows the programmer to override the standard plan selection mechanism. A meta-level plan was defined that used a reinforcement learning algorithm to select plans, with the agent being rewarded or penalised when it reached certain states.

The main criteria for the reinforcement learning algorithm was that it had to be an unsupervised learning algorithm — the agent had to learn from its mistakes based on the feedback from the world, rather than external feedback. The algorithm that was selected was Q-learning [17], a well-studied learning algorithm. This algorithm is by no means the only candidate for the task (e.g. Sarsa would be equally applicable), and different learning algorithms do produce some variation in behaviour (as do the parameter settings of any given algorithm) [16]. However human learning is different again to any of these machine learning algorithms, and it was felt that Q-learning provided as good an approximation as any other.

In the Q-learning algorithm, the state is characterised by a limited number of parameters, and each state-action pair has a Q-value, which is constantly updated at run time. The update equation is defined by

$$\begin{aligned} Q(s_t, a_t) \quad \leftarrow \quad & Q(s_t, a_t) + \\ & \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]. \end{aligned}$$

α is the learning rate of the agent, and γ is the payoff — a discount on the expected reward when selecting the 'best' next option.

One of the problems with using this algorithm is that it introduces a range of parameters to the agent (α, γ and the reward function) which must then be set. The point of using a folk psychological explanation for the extension was to maintain the 'naturalness' of the representation, but by introducing a reinforcement learning mechanism in the implementation of it, artificial constructs have been introduced. Fortunately the initial experiments, described below, showed that the agent was not particularly sensitive to these parameters, and default values would generally be acceptable.

Another difficulty is in representing the world in terms of states. In the initial experiment described below, this was straight-forward, but in general, the environments in which we wish to situate human models are complex ones, and if we were to use the raw data from these environments to represent the state space, it would be too large to be manageable. A surprisingly simple solution arose to deal with this problem in the early stages of the human modelling experiments, as described in Section 4.4.

4.3. Initial Experimental Results

The initial experimental testbed was a simple pathfinding task in a gridworld, as shown in Figure 2. The aim is for the agent to find a path from any point on the grid to a goal (green) square, while avoiding the anti-goal (red) squares. The configuration shown here is a sample one — the grid can have any combination of penalty or reward squares. Moreover, it is possible to change this configuration at run-time, and observe how the agent behaves in a dynamic environment. The agent's goal is to find a shortest path to the goal square, from any point on the map. The task itself is modelled upon one used by Masson to demonstrate Q-learning [8].

Upon creation of the agent, it is placed at a random position in the world, and it then 'walks a path' until it reaches a goal or anti-goal square. By default, it is only rewarded when it reaches a goal square and penalised when it goes through a penalty square, but it is also possible to penalise it on normal squares. At the completion of a path, the agent is randomly repositioned in the world, whereupon it repeats the process. The agent uses an ϵ-greedy exploration algorithm, by default using random exploration 10% of the time, but otherwise selecting the plan with the highest Q-value.

For this task, agent itself is very simple, with one top level plan (`find-goal`), and four low level plans (`move-north`, `-east`, `-south`, `-west`). Conceptually, the agent learns to recognise the best move for a particular position on the grid. This simple test case provided the ability to test the sensitivity of the agent to the parameters. Three different maps were used: as well as that shown in Fig. 2 (referred to as map C), a simple map with the four central squares set as goals was used (map A), and one with a goal at (5,5) and an anti-goal at (6,6) (map B). The measure of effectiveness that was used was the proportion of states for which the maximal Q-value represented an optimum move. Figure 3 presents the effects of some of the variations of parameters in the different maps.

It can be seen from Fig. 3(a) that variations in the magnitude of the penalty or reward applied for goals and anti-goals (even by a factor of 100) has very little impact on the agent, but penalising the agent for making a move (even by a relatively small amount) decreases the performance in the early stages, even though it significantly increases it in the long term. If the aim is to reach a 'reasonable' level of accuracy as quickly as possible, penalising a move that does not actually result in a 'bad' state is obviously a counter-effective strategy. When one considers the remaining parameters, adjustments seem to have little effect on the performance in early stages; it is only in the long term that they have an impact on performance, and even then, not by much. Figure 3(b) then contrasts some of the results for the three maps. These are typical in that however the parameters were varied, the results for map C followed the trend of the other two maps, but more slowly.

As Masson explains in the description of his Q-learning demo, "it would be silly to build a pathfinder using Q-learning" [8]. For this problem there are specifically-tailored techniques that are more appropriate. However this somewhat artificial demo does allow exploration of the parameters, and has some correspondence to the real world decision problems that this extension is designed to deal with. Firstly, the number of options at any point is about right: faced with a particular goal, a person typically does not consider a wide range of possible options, usually no more than three or four. Secondly, the idea of there being certain 'good' states, certain 'bad' states, but a large number of indeterminate states is also true of real-world problems. Moving one step closer to a goal square is equivalent to completing a sub-plan in a real-world problem: it might get you

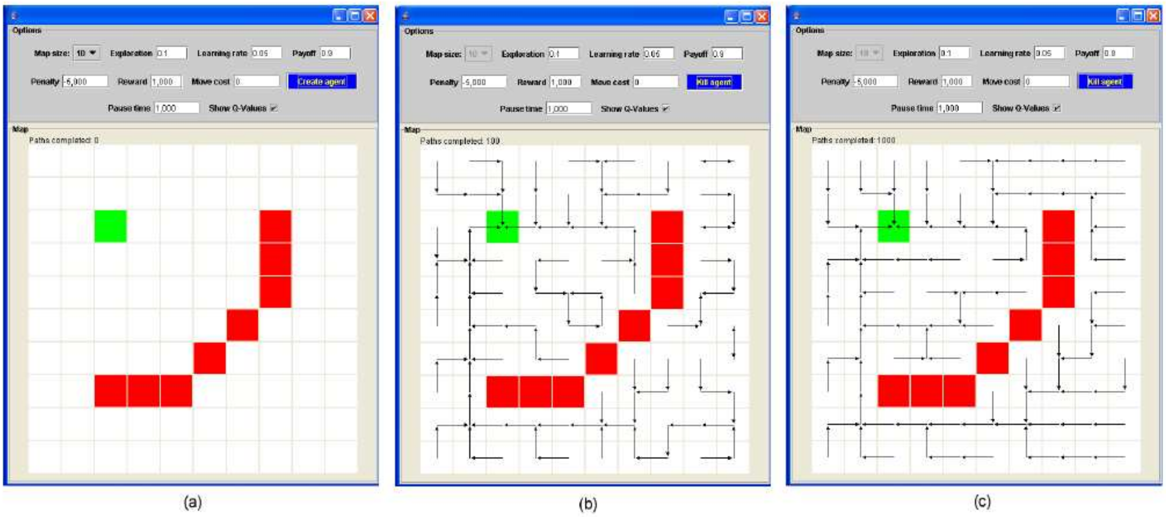

Figure 2. The path-finding task (map C): (a) initial state (b) after 100 iterations (c) after 1000 iterations

that little bit closer to the ultimate goal, but it is usually fairly insignificant in the scheme of things.

4.4. Applying the extension to a 'real' problem

The second experiment was designed to test the extension in an environment which displayed the characteristics that typically lead to this style of decision making being used. This environment was a first-person shooting game, Quake2.

Three expert players of Quake2 were selected, who would serve as subjects to be modelled. Originally, the aim was to build two models of each of the players, one using 'standard' JACK, and the other using the RPD-enhanced JACK, and then recruit other Quake2 players to play against these models and judge if the enhancement made the models seem more human-like. This lofty aim was modified somewhat, due to difficulties in interfacing with the game engine.

While it was relatively simple to connect the agents to the engine, to have them recognise players and items and run around, comprehending the map was a different matter. The map in Quake2 is represented in a polygon-based structure, the elements of which are used to render the walls, doorways, stairs, et cetera on the user's screen. While it was relatively simple to use this structure to do A* path planning, it was far from easy to recognise features of the landscape. If the experts being modelled had ignored these features, it would not have been a problem, but all too often, they used plans like "*If I'm being chased and running through a doorway, I fire a grenade at the wall above the doorway, because it will bounce off on the person chasing me.*" Representing a plan such as this is not difficult, however due to the interface problems it never gets used, because the agent is never able to 'see' a doorway. The Quake-playing agents *do* adapt their behaviours (they can learn, for example, to run from a fight if a third combatant appears). However much of the experts' behaviour referenced map features that the agents could not see, and so the agents behaviour turned out to be such a poor approximation to human behaviour that asking if they were 'more' human-like when using RPD was not a meaningful question.

Two important results did emerge from the building of these models though, the first with respect to BDI-based human modelling, and the other specific to the RPD enhancement. The general result relates to the issue of knowledge elicitation. Knowledge elicitation is one of the more difficult and time-consuming stages of human modelling, particularly when the task being modelled is a complex one. Ensuring that the knowledge is interpreted correctly, and that *all* the relevant knowledge is collected is non-trivial. For the models of Quake2 players, a methodology called applied cognitive task analysis [10] was adapted to gather the knowledge to program the JACK agents, as described in [14].

The second result was an answer to the problem of managing the size of the state space, as was briefly mentioned in Section 4.2. In the Quake2 domain, if the raw data from the game engine was used, the size of the state space would be infeasibly large. Even just taking the position variable, the number of possibilities was orders of magnitude larger than the gridworld example. To start with, position was specified in three dimensions, and each dimension using a real, not integer, number. Then on top of position, there was health,

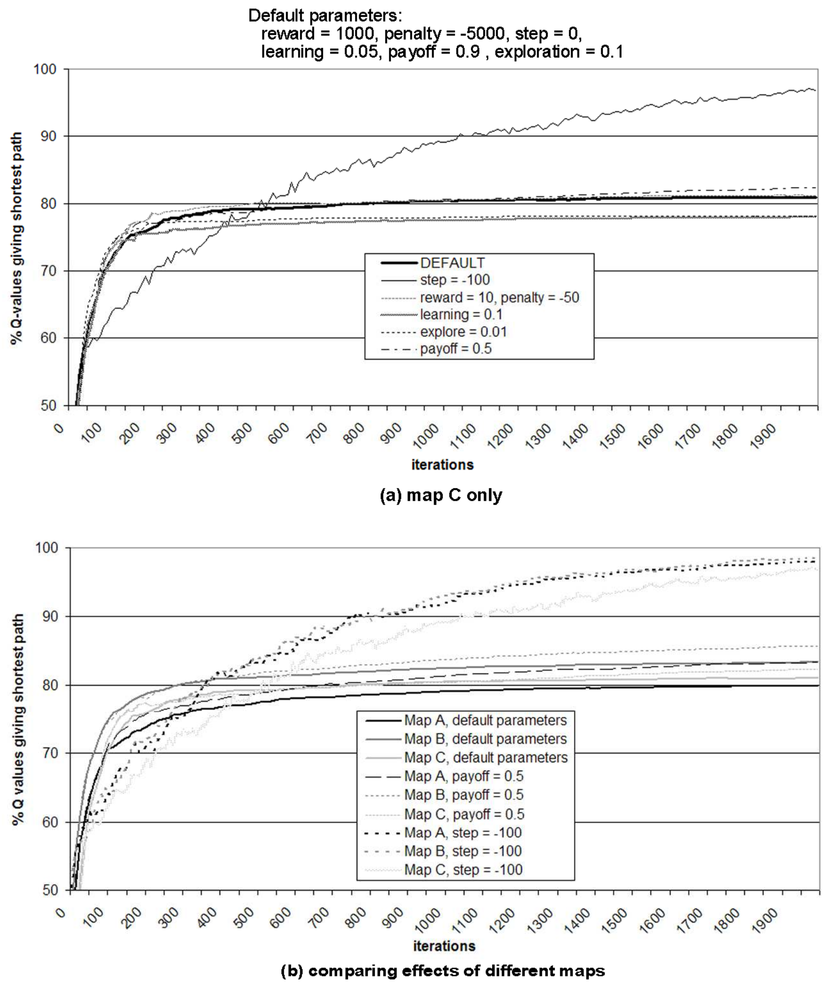

Figure 3. The effects of varying parameter settings

weapon, ammunition, armour, opponent,... the list went on. Obviously it would be impossible to use Q-learning if the space was represented in terms of the distinct values of these variables. What emerged from early discussions with the experts though is that (of course) they themselves abstract these variables into higher-level constructs, such as "*open area*," as opposed to "*position (x,y,z)*" or "*poor health*" instead of "*health = 30*". By using these abstractions to characterise the world, it was possible to reduce the state space to 100-200 distinct possibilities, depending on the player being modelled.

5. The Limitations of Folk Psychology

The limitation of this approach is that while there are folk psychological explanations of many aspects of human reasoning, such as the RPD model of decision making described in Section 4, or Ortony, Clore and Collin's model of emotions [15], some aspects of human behaviour do not have suitable folk psychological explanations. Folk psychology 'bottoms-out' at a certain level, failing to account for some of the low level details of human behaviour, for example those related to physical and mental reaction times. For many applications, this will not matter: a model based in folk psychology may provide as much of the detail of human behaviour as is required. However for those applications that do require these low level details to be included, another approach is required.

One approach that has been used to model perception and action is to extend the BDI framework with 'external' models of perception and action, rather than integrating them into the framework [13]. The perception and action models constrain the reasoning of the agent, because it must wait for actions and eye movements to occur, and correct for errors in movement, but they do not directly influence internal mechanisms in the BDI framework. It is not clear that this approach would be suitable for *all* characteristics of human behaviour that cannot be modelled with folk psychology, but it is a starting point.

6. Conclusions

The BDI framework has been used with considerable success for human modelling, but was not designed specifically for this purpose. A framework that included more of the basic characteristics of human behaviour would be a more powerful modelling tool, because the model builders could the focus on the domain-specific aspects of the model being built, reducing their overall workload. However the BDI framework does have one particular strength for human modelling — its folk psychological roots — which in itself facilitates the model builders' task, because the subjects' knowledge maps easily to the constructs used in the framework. This paper has presented an approach to developing a framework that represents more of the basic human characteristics, but maintains this conceptual simplicity.

The approach described is an incremental one, successively adding support for additional characteristics, based upon folk psychological models of these characteristics. As was demonstrated with the implemented example, even relatively simple folk psychological explanations *may* add 'unnatural' parameters to the framework, and this needs to be taken into account when evaluating the advantages of any addition. It is also noted that folk psychological explanations of different characteristics sometimes use variations in the basic concepts, and the mapping between concepts should be carefully considered.

There are many human characteristics that could be incorporated into the framework using this approach, but some aspects of human behaviour and reasoning do not lend themselves to folk psychological explanations, and this sets the bounds of the approach. If these types of characteristics are to be included in the framework, they must be incorporated in some other way, as with the models of perception and action described in Section 5.

This approach to extending the BDI framework should produce a framework that provides better support for human modelling, by maintaining the strength of the BDI architecture, but addressing some of its weaknesses. While it will not be possible to add models of *all* basic human characteristics using this approach, many of those that are seen as important in human modelling can be handled this way. A complementary approach can address those characteristics that do not have folk psychological explanations.

References

[1] Agent Oriented Software Pty. Ltd. JACK Intelligent Agents. http://www.agent-software.com/shared/products/index.html.

[2] M. E. Bratman. *Intention, Plans and Practical Reason*. Harvard University Press, Cambridge, Massachusets, 1987.

[3] D. C. Dennett. *The Intentional Stance*. The MIT Press, Cambridge, Massachusets, 1987.

[4] C. Heinze, S. Goss, T. Josefsson, K. Bennett, S. Waugh, I. Lloyd, G. Murray, and J. Oldfield. Interchanging agents and humans in military simulation. In *Proceedings of the Twelth Innovative Applications of Artificial Intelligence Conference*, 2001.

[5] G. Klein. *Sources of Power*. The MIT Press, 1998.

[6] G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsambok, editors. *Decision Making in Action: Models and Methods*. Ablex Publishing Corporation, 1993.

[7] F. Lui, D. Jarvis, J. Jarvis, R. Connell, and J. Vaughan. An architecture to support autonomous command agents for onesaf testbed simulations. In *Proceedings of the SimTecT Conference 2002*, Melbourne, Australia, 2002.

[8] T. Masson. Artificial intelligence – the Q-learning algorithm. http://thierry.masson.free.fr/IA/en/qlearning_about.htm.

[9] D. McIlroy and C. Heinze. Air combat tactics implementation in the Smart Whole AiR Mission Model (SWARMM). In *Proceedings of the First International SimTecT Conference*, Melbourne, Australia, 1996.

[10] L. G. Militello and R. J. B. Hutton. Applied Cognitive Task Analysis (ACTA): A practitioner's toolkit for understanding cognitive task demands. *Ergonomics*, 41:1618–1641, 1998.

[11] P. Molyneux. Postmortem: Lionhead Studios' Black & White. *Game Developer*, June 2001.

[12] G. Murray, S. Steuart, D. Appla, D. McIlroy, C. Heinze, M. Cross, A. Chandran, R. Raszka, G. Tidhar, A. Rao, A. Pegler, D. Morley, and P. Busetta. The challenge of whole air mission modelling. In *Proceedings of the Australian Joint Conference on Artificial Intelligence*, Melbourne, Australia, 1995.

[13] E. Norling and F. E. Ritter. Embodying the JACK agent architecture. In M. Stumptner, D. Corbett, and M. Brooks, editors, *AI 2001: Advances in Artificial Intelligence*, volume 2256 of *Springer Lecture Notes in Artificial Intelligence*, pages 368–377. Springer, 2001.

[14] E. Norling and L. Sonenberg. Creating interactive characters with BDI agents. In *Proceedings of the Australian Workshop on Interactive Entertainment IE2004*, Sydney, Australia, February 2004.

[15] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.

[16] K. Takadama, Y. L. Suematsu, N. Sugimoto, N. E. Nawa, and K. Shimohara. Cross-element validation in multiagent-based simulation: Switching learning mechanisms in agents. *Journal of Artificial Societies and Social Simulation*, 6(4), 2003. http://jasss.soc.surrey.ac.uk/6/4/6.html.

[17] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.