

Cognitive Hybrid Reasoning Intelligent Agent System

Christos Sioutis¹ and Nikhil Ichalkaranje¹

School of Electrical and Information Engineering,
Knowledge Based Intelligent Engineering Systems Centre,
University of South Australia, Mawson Lakes SA 5095, Australia
{Christos.Sioutis,Nikhil.Ichalkaranje}@unisa.edu.au
<http://www.kes.unisa.edu.au>

Abstract. This paper presents a novel approach for implementing reasoning and learning systems based on Intelligent Agents. It is particularly suited for agent systems working in complex and cooperative dynamic environments. This approach was developed to overcome limitations confronted while researching the use of agents in such environments. A brief background of the underlying conceptual reasoning model is described, followed by details of the implemented framework and concluded with some final remarks and future work.

1 Introduction

Recent research directions [9, 7] for enhancing the performance of agent based systems have lead to the investigation of human reasoning models as a main paradigm for implementing intelligent agents. The widely used Beliefs Desires and Intentions (BDI) model was developed by Bratman [2] to describe how humans perform practical reasoning. Practical reasoning is defined as considering actions that need to be performed given a particular situation [10]. An important assumption about the BDI model is that a BDI agent's actions must be logically consistent with the combination of its beliefs and goals. The BDI reasoning process is believed to consist of at least two distinct activities, the first activity being deliberation, which involves deciding what goals to achieve [10], and the second activity, means-ends reasoning, involves deciding how to achieve this goal [10].

JACK is a commercial platform for developing intelligent agents based on the BDI model that uses events and plans to define different behaviours that an agent can exhibit [1]. A behaviour is executed by posting an event which is handled by a plan. The plan contains any relevant code that causes the agent to exhibit the desired behaviour. A plan is also able to post other events and the JACK language provides a number of ways in which events can be posted and handled. This makes it a powerful platform for building complex agent behaviours. However, one limitation that was observed is that all plans must be pre-defined to be used. There is no way in the current distribution (version 5.0)

to modify a behaviour, short of modifying the plan code (or the way the plan is selected) and re-compiling it off-line. This limitation is specifically addressed by this research.

Section 2 presents a brief overview of a conceptual reasoning and learning model previously developed and ties our Cognitive Hybrid Reasoning Intelligent Agent System. Our case study related to a learning agent is presented in section 3. The final section presents concluding remarks and future directions.

2 Cognitive Hybrid Reasoning Intelligent Agent System

Rasmussen's decision ladder [5], Boyd's OODA loop [4] and Bratman's BDI are three popular models for describing human decision making. Recent research has also revealed that these models are complementary and can in-fact be fused together to yield a new, hybrid and more detailed conceptual reasoning model that supports learning [6]. The Cognitive Hybrid Reasoning Intelligent Agent System (*CHRIS*) is an implementation of this model that was developed as an extension to JACK. It provides a complete framework for designing agents based on the reasoning model. The framework however is not directly executable. A specific application must be designed such that it uses the framework and implements any required additional components. The only reason for this, is that any non-implemented components are inherently application specific, in-fact these components 'hook' the framework code with code from the specific application into one complete working system.

The framework divides the Agent's reasoning process into six stages. Each of the stages have been implemented in different Java packages which are required to be imported for successful compilation. Two additional packages are also required as listed below.

1. Agent: Provides an extended JACK learning agent, the CHRIS capability and associated Java interface classes.
2. Observation: Receives sensations and converts raw data into useable information.
3. Orientation: Uses new information to update the Agent's beliefs accordingly.
4. Decision: Manages goals that the agent is trying to achieve.
5. Action: Executes tasks that cause the agent take actions in the environment.
6. Learning: Modifies the actions taken by the agent according to specified learning goals, active and passive learning methods are available.
7. Logging: Controls the logging functionality built into the framework.
8. Util: Utility classes that were created while developing the framework.

Active learning is realised by traditional Reinforcement Learning (RL) algorithms. It involves handling a event that has an attached learning goal. The agent then activates the learning algorithm specified in the learning goal and executes different actions while observing the results of the actions. The learning is achieved by the algorithm finding an optimal state-action value function based on rewards generated by its learning goal. That is, a function that translates a

particular state to a specific action that is to be taken in that state. The optimal value function causes the agent to make choices that give it maximum rewards.

Passive learning is a feature that has been implemented specifically in the *CHRIS* framework but was inspired by the ‘exploration control module’ presented by Dixon [3]. It allows an RL action-selection policy to be replaced by a JACK plan written in a specific way. This allows an agent to learn the behaviour of a pre-written JACK plan, and evaluate the JACK plan with respect to the given learning goal. Skill control is another feature that allows the agent to switch from passive learning to active learning. When enabled, skill control uses specified thresholds for switching between the two learning methods.

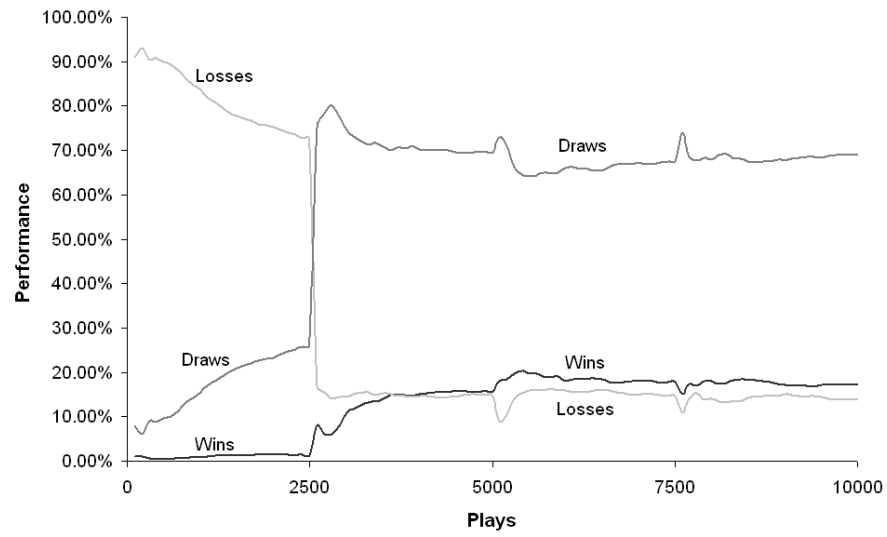
The interested readers are invited to contact the first author for a detailed description of *CHRIS*.

3 Example learning agent

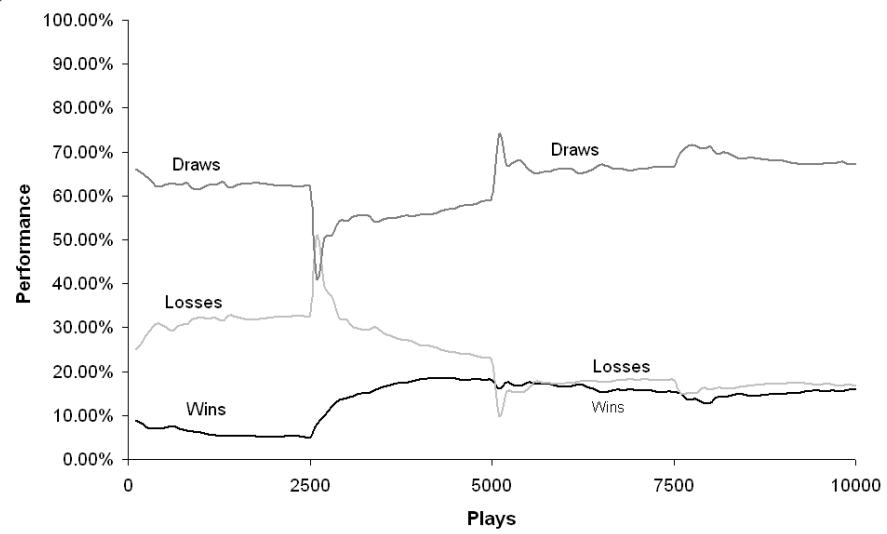
A simple example system was implemented in order to test the features provided by the framework, it involves an agent that learns how to play the game of Tic Tac Toe. The actions available to the agent are variable and are based upon the moves available to the agent for a particular state. The Sarsa [8] RL algorithm and the EGreedy [8] policy were selected. The learning goal’s reward function is based on the action that the agent chooses, it returns a -20 if the action results in the agent losing the game, a +20 if the action results in the agent winning the game and -1 if the action results in a stalemate or if the game is not finished. This instructs the agent to avoid losing and try to win in the quickest possible way. The required components for both active and passive learning were implemented and four runs of 2500 plays were made for each type of learning. The learned data was preserved after each run for that particular learning type to demonstrate that the agent is able to save, load and improve upon the value function over multiple executions.

Figure 1a shows the performance of the agent for active learning. The agent uses an exploration rate of 50% for the first run of 2500 plays. This is indicated by the very low performance of the agent during the first 2500 plays where it loses approximately 75%, draws approximately 25% and wins less than 1% of the games. The exploration rate is then set to 10% for the subsequent runs and the performance suddenly changes to approx 70% drawing, 15% winning and 15% losing. The small spikes in the lines every 2500 correspond to starting a new run.

Figure 1b shows the performance of the agent for passive learning. During the first 2500 runs the agent’s skill control is disabled. Which means that it learns from its actions but it acts based on a regular JACK plan that contains a behaviour for playing Tic Tac Toe, the learning package has no effect on the actions taken by the agent which performs at 65% draws, 35% losses and 5% wins. For subsequent runs the skill control is enabled which causes the agent to switch back and forth between active and passive learning depending on its performance. Again a substantial change in performance is achieved, draws remain



(a) Active Learning



(b) Passive Learning

Fig. 1: Learning Agent performance when playing Tic Tac Toe

at approximately 65% while both wins and losses stabilise at approximately 18%. This example demonstrates an advantage of passive learning. It performs well at the start when learning is inexperienced, but it also improves its performance later on when learning is so experienced that it provides a better behaviour than the pre-written JACK plan. The behaviour of the agent can be even further improved through tweaking its reward function or improving a state generalisation function defined within its learning goal.

4 Conclusions and Future Work

This paper describes a framework that is based upon a hybrid, conceptual reasoning model. The framework allows the creation of JACK agents with the ability to learn from their experiences. Learning is achieved through RL algorithms that execute different actions while observing the results of the actions until an optimal state-action value function is reached for a given learning goal. The optimal value function causes the agent to make choices that give it maximum rewards.

Future work for *CHRIS* includes implementing different types of learning agents and assessing their performance. Some additional RL algorithms (eg. Eligibility Traces) may also be implemented in order to improve learning performance. A Planning capability based on [8] is also intended to be integrated into *CHRIS*. The planning capability would allow the agent to learn a Model of the environment. The Model would be used to ‘mimic’ the environment in a limited way allowing the agent to do off-line learning using simulated experiences against the model.

5 Acknowledgements

The work presented in this paper is the product of research in a joint project with the University of South Australia and the Australian Defence Science and Technology Organisation. Specifically, team members who contributed significantly to this project are Pierre Urlings, Jeffrey Tweedale and Lakhmi Jain. In closing, it is acknowledged that significant assistance has been provided by Dennis Jarvis and Jacque Jarvis of Agent Oriented Software in the understanding and application of the JACK technology to the project.

References

1. Agent Oriented Software, *Jack intelligent agents agent manual*, [Online, accessed April 2005], (2005), URL:<http://www.agent-software.com.au/shared/resources/index.html>.
2. Bratman ME, *Intention, Plans, and Practical Reason*, Center for the Study of Language and Information, (1999).
3. Dixon KR, Malak RJ, and Khosla PK, *Incorporating Prior Knowledge and Previously Learned Information into Reinforcement Learning Agents*, Technical Report, Institute for Complex Engineered Systems, Carnegie Mellon University, (2000) pp13.

4. Hammond GT, *The Mind of War: John Boyd and American Security*, Smithsonian Institution Press, Washington, USA, (2004).
5. Rasmussen J, Pejtersen AM, Goodstein LP, *Cognitive Systems Engineering*, Wiley & Sons, New York, NY, (1994).
6. Sioutis C, Ichalkaranje N, Jain LC, Urlings P, and Tweedale J, *A conceptual reasoning and learning model for intelligent agents*, In proceedings of the 2nd International Conference on Artificial Intelligence in Science and Technology (AISAT 2004), University of Tasmania, Hobart, Australia, November, (2004) 301-307.
7. Sioutis C, Tweedale J, Urlings P, Ichalkaranje N and Jain LC 2004, *Teaming Humans and Agents in a Simulated World*, In proceedings of the 8th International Conference on Knowledge Based Intelligent Information and Engineering Systems (KES 2004), Springer Verlag, Berlin, pp. 80-86.
8. Sutton RS and Barto AG, *Reinforcement Learning An Introduction*, The MIT Press, London (2000).
9. Urlings P., *Teaming Human and Machine*, PhD Thesis, University of South Australia (2004).
10. Wooldridge M, *Reasoning About Rational Agents*. The MIT Press, London, (2000).