Computational methods for modeling human-like agents in virtual environment

Kang, Yilin

2014

Kang, Y. (2014). Computational methods for modeling human-like agents in virtual environment. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/62192>

<https://doi.org/10.32657/10356/62192>

COMPUTATIONAL METHODS FOR MODELING HUMAN-LIKE AGENTS IN VIRTUAL ENVIRONMENT

A thesis
Submitted to the School of Computer Engineering
Nanyang Technological University

by

Kang Yilin

in partial fulfilment of the requirement for the
degree of Doctor of Philosophy

August, 2014

Abstract

Virtual environments, also known as virtual worlds, are computer simulated spaces which enable players or participants to engage in long-term, joint coordinated actions, with the novel possibilities which may be impossible or too costly in a real world setting. Hence, they have become a popular platform used in a variety of contexts, including teaching in classrooms, informal learning, distance learning, business, and e-commerce.

In a virtual world, the primary population is constituted by real people, represented as human-controlled agents or avatars. Besides that, computer-controlled agents, also known as NPCs (non-player characters), play an important role. Given the NPC's popularity in virtual worlds, many people have been working on various approaches to incorporate intelligent learning agents to improve the interactivity and playability.

Three key requirements of realistic characters in virtual worlds have been identified, namely (1) Autonomy: Agents should be able to function autonomously and make decisions proactively in response to events happening in the virtual world around them; (2) Interactivity: The interaction between virtual humans and real humans should be as natural as possible, integrating both verbal and nonverbal communication such as facial expressions and body gestures; and (3) Personality: Virtual humans should exhibit human-like traits and characteristics, such as personalities and emotions. As far as we know, the existing work mainly focuses on the different building parts of the virtual human, such as animation, body language, natural language processing, emotion, decision making, learning and so on. We propose an architecture that integrates autonomy, interactivity and personification, which are identified as the three most important features for virtual humans. No one has proposed a model that emphasizes such features. This is the motivation for our work.

The first part of the dissertation presents a self-organizing neural model for creating intelligent learning agents in virtual worlds. As agents in a virtual world roam, interact

and socialize with users and other agents as in real world without explicit goals and teachers, learning in virtual world presents many challenges not found in typical machine learning benchmarks. In this dissertation, we highlight the unique issues and challenges of building learning agents in virtual world using reinforcement learning. Specifically, a self-organizing neural model, named TD-FALCON (Temporal Difference - Fusion Architecture for Learning and Cognition) is deployed, which enables an autonomous agent to adapt and function in a dynamic environment with immediate as well as delayed evaluative feedback signals. Specifically, TD-FALCON integrates temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. By incorporating TD-FALCON, an agent will be able to learn from sensory and evaluative feedback signals received from the virtual environment without involving human supervision and intervention. In this way, the agent needs neither an explicit teacher nor a perfect model to learn from. Performing reinforcement learning in real time, it is also able to adapt itself to the variations in the virtual environment and changes in the user behavior patterns. Furthermore, by incorporating temporal difference learning, TD-FALCON agents can overcome the issues, such as the absence of immediate reward (or penalty) signals in virtual world by estimating the credit of an action based on what it will lead to eventually. We have implemented and evaluated TD-FALCON agents as tour guides in a virtual world environment. Our experimental results show that the agents are able to adapt and improve their performance in real time.

To enable more dynamic and personalized experience, we propose to incorporate adaptive user modeling with our learning agents. Specifically, a two channel fusion ART [1, 2], is employed for learning player models through the interaction with the players. By formulating cognitive codes associating agent’s recommendation to user’s feedback, fusion ART learns explicit player models of the users’ likes and dislikes. We have developed personal agents with adaptive player models as tour guides in a virtual world environment. Our experimental results show that the agents are able to learn user models that evolve and adapt with players in real time. Furthermore, the virtual tour guides with player models outperform those without adaptive player modeling.

Working towards the challenges of creating realistic characters, this dissertation proposes a brain inspired agent architecture that integrates goal-directed autonomy, natural

language interaction and human-like personification. Based on self-organizing neural models, the agent architecture maintains explicit mental representation of desires, personalities, situation awareness and user awareness. Autonomous behaviors are generated via evaluating the current situation with active goals and learning the most appropriate social or goal-directed rule from the available knowledge, in accordance with the personality of each individual agent. We have built and deployed realistic agents in an interactive 3D virtual environment. Through an empirical user study, the results show that the agents are able to exhibit realistic human-like behavior, in terms of actions and interaction with the users, and are able to improve user experience in virtual environment.

Keywords: Human-Like Characters, Self-Organizing Neural Networks, Cognitive Agents, Adaptive Learning, User Model.

Acknowledgments

First and foremost, I'd like to express my sincerest gratitude to my supervisor, Dr. Tan Ah-Hwee, who has guided and supported me throughout my Ph.D. candidature with his knowledge and patience. I have not only acquired in-depth knowledge in my research area but also learnt to be a respectful person from him as my role model. I am also thankful for countless time and effort he spent on helping me improve my research skills. One simply could not wish for a better or friendlier supervisor.

Special thanks go to Dr. Fiona Nah. She introduced me the way of evaluating a system systematically. Without her help, I would have spent much more time in finishing this research work. I am indebted for her precious time for discussing with me over research issues and being supportive all the way here.

I'd like to thank Dr. Miao Chunyan for her kindness help in giving me many precious advices on finishing this manuscript. The award-winning thesis she referred to me gives me the ideas of how to write this manuscript in a clear way.

I'd also like to extend my appreciation to Dr. Budhitama Subagdja, Dr. Wang Di and Dr. Jiang Xing. They have helped me in many ways in the preparation and completion of this research work.

During my research, Mr. Jijun and Mr. Benny have helped me a lot on project implementation and management. I'd like to give my sincere thanks to them for their generous help.

I would like to thank my parents and my husband, for supporting me throughout all my studies. Their dedication, love and persistent confidence in me, are the warmest support I've ever received. Last but not the least, I'm very thankful for having my son Zikang during my PhD study. His cute and smile always encourage me to overcome the difficulties in life. I own every achievement to all of them.

Contents

Abstract	i
Acknowledgments	iv
List of Figures	x
List of Tables	xiii
1 Introduction	1
1.1 Background and Motivation	2
1.2 Issues and Challenges	3
1.2.1 Cognitive Agent model for virtual human	4
1.2.2 Learning in Virtual World	5
1.2.3 Enabling Personalized Virtual World Experience	6
1.3 Approach and Methodology	7
1.4 Contributions	8
1.4.1 Reinforcement Learning in Virtual Worlds	8
1.4.2 Learning Personal Agents with Adaptive Player Modeling	8
1.4.3 Self-Organizing Cognitive Models for Virtual Agents	9
1.4.4 Case Study and Evaluation of Agent based Virtual Human	9
1.5 Thesis Organization	10
2 Literature Review	12
2.1 Virtual Agents	12
2.1.1 Rule-Based Systems	12
2.1.2 Finite State Machine	13
2.1.3 Fuzzy Logic	14
2.1.4 Decision Trees	15

2.1.5	Neural Networks	16
2.1.6	Evolutionary Agents	17
2.1.7	Bayesian Learning	17
2.1.8	Other Approaches	18
2.2	Player Modeling	19
2.2.1	Interactive Storytelling	19
2.2.2	Poker Games	19
2.2.3	Other game genre	20
2.3	Modeling Virtual Human	20
2.4	Cognitive Models	22
2.4.1	State, Operator, and Result (Soar)	22
2.4.2	Adaptive Control of Thought-rational (ACT-R)	22
2.4.3	ICARUS	23
2.4.4	Belief-desire-intention (BDI)	25
2.4.5	The Subsumption Architecture	26
2.4.6	Connectionist Learning with Adaptive Rule Induction On-line (CLAR- ION)	27
2.5	Evaluation of Human-like Agents	29
2.6	Summary	30
3	Towards Building Human-Like Virtual Agents	31
3.1	Introduction	31
3.2	Properties of Virtual Human	32
3.2.1	Autonomy	32
3.2.2	Interactivity	33
3.2.3	Personification	34
3.3	Preliminaries	34
3.4	A Framework for Human-Like Agent Models	35
3.4.1	Sensory-Motor Layer	37
3.4.2	Mental Layer	37
3.4.3	Cognitive Agent Model Formulation	38
3.5	Summary	39

4	Reinforcement Learning in Virtual Worlds	40
4.1	Introduction	40
4.2	Problem Formulation	41
4.3	TD-FALCON as Virtual Agents	42
4.3.1	Self-Adaptation	43
4.3.2	Generalization	43
4.3.3	TD Learning	43
4.3.4	Fast and Stable Real-time Learning	43
4.4	Learning Algorithm	44
4.4.1	FALCON Dynamic	44
4.4.2	TD-FALCON	46
4.5	A Case Study on YOV Co-Space	47
4.5.1	Embodiment of TD-FALCON in Co-Space	48
4.5.2	Empirical Experiments	50
4.5.3	User case study	57
4.6	Summary	58
5	Learning Personal Agents with Adaptive Player Models	60
5.1	Introduction	60
5.2	Problem Formulation	61
5.3	Fusion ART	61
5.4	A Case Study on Learning Player Models in YOV Co-Space	64
5.4.1	Modeling Players	64
5.4.2	Learning Player Models	64
5.4.3	Input Space Representation	66
5.4.4	Output Space Representation	66
5.5	Integrating Player Model and Personal Agent	67
5.6	Empirical Experiments	67
5.6.1	Experimenting with Static Player Profile	69
5.6.2	Experimenting with Evolving Player Profile	70
5.6.3	Experimenting without Player Registered Profile	71
5.7	Summary	72

6	Self-Organizing Cognitive Models for Virtual Human	73
6.1	Introduction	73
6.2	A Neural Network Instantiation of Virtual Humans	73
6.2.1	Architecture	73
6.2.2	Network Dynamics	76
6.3	Natural Language Understanding and Natural Language Generation . . .	78
6.3.1	Existing Work	78
6.3.2	Extending Cognitive Agent Models	80
6.4	System Properties	82
6.4.1	Human-Like Properties	82
6.4.2	Further Discussion	84
6.5	Summary	85
7	Case Study and Evaluation of Agent based Virtual Human	86
7.1	Introduction	86
7.2	A Case Study on NTU Co-Space	87
7.2.1	Embodiment of Agent Architecture	88
7.3	Theoretical Background on User Experience	90
7.3.1	Definition of UX	90
7.3.2	Related Constructs	92
7.4	Evaluative Experiments	94
7.4.1	Research Methodology	94
7.4.2	Data Analysis	98
7.5	Summary	101
8	Conclusions and Future Work	102
8.1	Summary of Contributions	102
8.1.1	Framework for Cognitive Agents	102
8.1.2	Reinforcement Learning in Virtual Worlds	102
8.1.3	Learning Personal Agents with Adaptive Player	103
8.1.4	Self-Organizing Cognitive Models for Virtual Agents	103
8.1.5	Evaluation of Agent based Virtual Human	103

8.2	Future Work	104
8.2.1	Adaptive Player Modeling	104
8.2.2	Extending The Cognitive Models for Virtual Human	105
8.2.3	Extending The Evaluation of Agent Based Virtual Human	106
A	References for Measurement Scales	107
B	Results of Principle Component Analysis	109
	List of Author's Publications	111
	References	113

List of Figures

1.1	Virtual characters in this dissertation.	2
1.2	The issues and challenges that this dissertation focuses on.	4
1.3	The flow of this dissertation.	8
2.1	Generic finite state machine diagram	14
2.2	Overview of fuzzy logic process	15
2.3	An example of decision trees	16
2.4	An example of artificial neural network	17
2.5	Overview of Evolutionary process	18
2.6	The Soar Cognitive Architecture.	23
2.7	The ACT-R 5.0 model.	24
2.8	The schematic diagram of ICARUS.	25
2.9	The BDI architecture in dMARS specification.	26
2.10	The subsumption architecture.	27
2.11	The CLARION architecture.	28
3.1	A schematic diagram of the cognitive agent model.	36
4.1	An agent interacting with the environment	41
4.2	TD-FALCON architecture.	42
4.3	The architecture of Co-Space.	48
4.4	A screenshot of TD-FALCON based personal agent in Co-Space.	49
4.5	Recommendation accuracy of TD-FALCON with and without pre-inserted rules.	52
4.6	User satisfaction index obtained over time.	53

4.7	User satisfaction index with and without pre-inserted rules.	54
4.8	Recommendation accuracy of TD-FALCON and the rule based agent (RBS).	55
4.9	User satisfaction index obtained by TD-FALCON and the rule based system (RBS).	56
4.10	Average recommendation accuracy (with range bars) of TD-FALCON agents without immediate rewards.	57
4.11	User satisfaction index of TD-FALCON agents without immediate rewards.	58
5.1	The architecture of fusion ART.	62
5.2	A two channel fusion ART architecture for supervised learning of user models.	65
5.3	The architecture of recommendation agent.	68
5.4	Recommendation accuracy of recommendation agents with registered player profile.	70
5.5	Recommendation accuracy of recommendation agent with evolving player profile.	71
5.6	Recommendation accuracy of recommendation agent without registered player profile.	72
6.1	A neural network implementation of the integrated cognitive agent model.	74
6.2	A typical pattern matching approach for dialogue generation.	79
6.3	The detailed implementation of the integrated cognitive agent model.	81
6.4	The process of natural language understanding and generation.	82
7.1	System architecture of NTU Co-Space.	88
7.2	A player touring the NTU Co-Space.	89
7.3	An example of agent profile model.	89
7.4	A screenshot of the static NPC in E1.	95
7.5	A screenshot of the AIML based NPC in E2.	95
7.6	A screenshot of the fusion ART-based agent in E3.	96
8.1	Proposed theoretical model and research hypotheses.	106
B.1	Factor analysis of constructs	109

B.2	Factor analysis of constructs cont.	109
B.3	Factor analysis of constructs cont.	110
B.4	Factor analysis of constructs cont.	110

List of Tables

2.1	Comparison of the representative virtual humans.	21
4.1	Attributes and possible values in the state space	49
4.2	User logic for generating rewards.	51
4.3	TD-FALCON parameter setting.	52
4.4	Performance of TD-FALCON with different parameter value settings. . .	53
4.5	A sample set of pre-inserted rules.	54
4.6	Result of real user study.	58
5.1	Attributes and possible values in the state space of player model.	65
5.2	Logic for generating rewards based on a synthetic player model P.	69
5.3	Fusion ART Parameter setting for Player Models.	69
6.1	Example of a typical pattern matching mechanism	79
7.1	Knowledge of campus.	88
7.2	A sample interaction rule.	90
7.3	A sample social rule.	90
7.4	Attributes and possible values in the state space.	91
7.5	Categories and attributes in the action space.	91
7.6	The overall performance of the three environments.	98
7.7	Measurement of the 3D virtual world skill under the three environments.	99
7.8	Descriptive Statistics of the six constructs measured in the three environ- ments.	100
7.9	F-test results of the six constructs under three types of comparison. . . .	100
A.1	References for Measurement Scales	107

A.2	References for Measurement Scales cont.	108
-----	---	-----

Chapter 1

Introduction

Virtual environments, also known as virtual worlds, are computer simulated spaces which enables players or participants to engage in long-term, joint coordinated actions. Other than predefined narrative applications, in these spaces, cultures and meanings come from a complex set of interactions [3]. Moreover, it provides people the novel possibilities which may be too costly in a real world setting. Hence, it has become a popular platform used in a variety of contexts, including teaching and learning(e.g. virtual classrooms in Open Sim), e-commerce (e.g. virtual retail stores in Second Life) , computer games (e.g. World of Warcraft), simulation of life (e.g. The Sims, Entropia Universe) and virtual military training (e.g. Virtual Battle Space). As the use of such worlds becomes more and more widespread, studies in South Korea have shown that users prefer virtual world to television [4].

In view of the development history of virtual worlds, other than the dramatic improvement in simulation and visualization given the emergence of new computing platforms based on highly parallel, many-core processors for both graphics and general-purpose computing, virtual worlds have gradually employed agent based characters as part of their inhabitants rather than all human controlled avatars. Moreover, more and more agent based characters are embodied with artificial intelligence in some degree. Their roles are various which depend on the nature of the given world, such as serving as elements of the worlds setting [5], being used for research purposes [6] and so on. In this dissertation, some of the main issues towards building human-like virtual agents have been investigated, and as shown in Figure 1.1, virtual characters with different features

are created. Moreover, a systematic evaluation of how they improve user experience in virtual worlds has been explored.

Figure 1.1: Virtual characters in this dissertation.

1.1 Background and Motivation

In a virtual world, the primary population is constituted by real people, represented as human-controlled agents or avatars. The advantage of it is that instead of altering the environment in significant ways to keep them interested, the virtual worlds' constant changes and growth come from the players themselves. Play can vary from activities such as sky diving, navigating and teleporting to the social experiences of creating cities, holding conferences and opening a market place. In this way, if the virtual world itself lacks a clear direction and a definite end, the players would still keep it not boring.

However, the potentially disadvantage of this kind of virtual worlds is that players, for all their sociability and dynamism, could rarely fit into the games atmosphere and goals as the designers carefully crafted [7]. Unlike in single-player game which players must follow the game themes, in virtual world, multiple players would most probably alter the designers' intentions. Given these reasons, in recent years, computer-controlled agents, also known as NPCs (non-player characters) have played an important role in virtual world as well [8].

We believe that the human-like agents are essential to virtual worlds for two reasons. First, ideally players themselves are willing to take on all the roles the society needs,

such as tour guides or personalized assistants. However, in reality, most player based characters prefer to do things that are difficult or impossible to do in real worlds. Hence, the supporting roles come to the agents' responsibilities. Secondly, agent-based characters could be designer's active representations. They could direct the players to those interesting, unusual, engaging parts of the environment. From the social level, the agents motivate the players' actions, in the form of adventure, quests, economics, legends and so on; From emotional level, the feeling of empathy could be created by the agents so that the players would be drawn into the storyline. Therefore, agents play significant important role to make the virtual world itself interesting to players.

Despite all these advantages, creating human-like agents requires a multidisciplinary effort, joining traditional artificial intelligence problems with a range of issues from computer graphics to social science [9]. Essentially, virtual humans must be able to act and react to the players as well as the virtual world itself. Hence, it needs reasoning and planning. To hold a conversation with players and other agents, speech recognition, natural language understanding, natural language generation and speech synthesis are all necessary. To control the virtual human's body in real time requires computer graphics and animation. To behave like a real human, social perspective is in need as well. As far as we know, the existing work has focused on these various aspects without building an architecture that integrates the three most important features of virtual humans.

In this dissertation, we would focus on the "mind" of virtual agents other than the appearance and animation. Specifically, learning ability, user modeling as well as the cognitive models for building virtual agents are discussed in the following chapters. We hope that through building an architecture that integrates the three most important features of virtual humans, other researchers or developers can use this architecture to build their own virtual human. They may also start from our architecture and do further exploration on how to integrate other features.

1.2 Issues and Challenges

As discussed in the previous section, the focus of this dissertation are three-fold, namely learning in virtual world, user modeling in virtual world and cognitive models for building virtual agents in virtual world. In fact, these three concepts couldn't be separated individually. Figure 1.2 shows the relationship between these three aspects.

Figure 1.2: The issues and challenges that this dissertation focuses on.

1.2.1 Cognitive Agent model for virtual human

Beyond learning and user modeling, greater challenges must be overcome to make these virtual characters convincingly realistic. First, they should be able to function autonomously within the environment they reside. To do so, they need to understand the context of the environment and the other virtual characters and human users co-existing in the environment. Also, the reasoning process for choosing the appropriate action should preferably not be scripted or predetermined. Instead, they should function autonomously and make decisions proactively in response to events happening in the virtual world around them. Second, the interaction between virtual humans and real humans should be as natural as possible, integrating both verbal and nonverbal communication such as facial expressions and body gestures. Third, virtual humans should exhibit human-like traits and characteristics, such as personalities and emotions. All of these require the convergence of a diverse range of technologies and domains, including cognitive science, natural language processing, decision making, dialogue management, speech and gesture generation and human-computer interaction which are complex to integrate.

1.2.2 Learning in Virtual World

With the popularity of virtual world, many people have been working on various approaches to incorporate intelligent learning agents to improve its interactivity and playability. Indeed, learning in a virtual world, just like in the real world, poses many challenges not addressed by traditional machine learning algorithms. In particular, learning in virtual world is typically unsupervised, without an explicit teacher to guide the agent in learning. Furthermore, it requires an interplay of a myriad of learning paradigms.

What Are The Goals? Traditional games, such as card games and board games, often have a clear objective of defeating the adversary. In first-person shooter games, the primary concern is to survive and kill the enemies. For a real-time strategy game, the goal is to secure the team and destroy the opponents' assets. In contrast, players in virtual world explore the environment, meet other residents, socialize, participate in activities, create and trade virtual properties and services with one another without explicit goals. This key characteristic has led to an important distinction between virtual world and traditional games on the requirement of learning.

The issue of implicit goals also brings about other issues such as how to measure the learning performance. As the players in the virtual world may roam in the virtual world without any keen desire, how we can measure the learning performance is one of the main concerns.

When to Start and Stop Learning? Exploring and playing in a virtual world is a continuous experience. Depending on the constraints and implementation of the virtual world, sensory signals and evaluative feedback are received and updated in synchronous or asynchronous fashion. It is thus important to associate an evaluative feedback to the correct state and action for reinforcement learning. Furthermore, agents need not learn all the time, as an agent may be left idle at times while the players wonder around or engage in chatting with other online residents. Enabling learning at all time consumes excessive computing resources and results in low efficiency.

How to Compute Evaluative Feedback? When comparing reinforcement learning with other types of learning methods, the most important feature distinguishing it from others is that it uses training information that evaluates the actions taken rather than the instructs of correct actions [10]. It requests for active exploration as well, which

includes an explicit trial-and-error search for good behavior [11]. In standard reinforcement learning tasks, the outcome of a trial is typically used in computing the evaluative feedback. In a First Person Shooting game, for example, a reward of 0.5 can be given for a hit and a reward of 1 is given for a kill during combat [12]. As virtual world is a continuous experience, evaluative feedback in virtual world cannot be defined simply by the final outcome of a game play. Since an agent is supposed to continue assisting or servicing a user, a more subtle way of defining evaluative feedback should be considered.

How to Learn in A Dynamic, Real-time Environment? Virtual world presents a highly dynamic environment. In particular, the rendering cycle of virtual world works at 50 or 60 Hertz so that changes appear as smooth animation [13]. Therefore, agents in virtual world cannot afford to operate in terms of primitive actions, such as “moves one step to north” or “turns to face northwest”. Instead, in order to allow learning to be conducted, abstractions have to be made so that forward searches could be done by learning in an abstract spaces in a manageable abstract space and then translate the solutions into action sequences in the original state space. In addition, learning should be made online in order to meet the real-time needs.

1.2.3 Enabling Personalized Virtual World Experience

There has been growing interest in creating intelligent agents in virtual worlds that do not follow fixed scripts predefined by the developers, but react accordingly based on actions performed by human players during their interaction. In order to achieve this objective, previous approaches have attempted to model the environment and the user’s context directly. However, a critical component for enabling personalized virtual world experience is missing, namely the capability to adapt over time to the cognitive, behavioral, and affective states of a particular player through the data derived from the interaction. User models are built on dynamic information obtained during virtual world interaction, but they could also rely on static player profiling information. So how to use computational methods to learn a user model and adapt the user model through the interaction with the users is a critical issue towards enabling personalized virtual world experience.

1.3 Approach and Methodology

In view of the issues and challenges in building realistic human-like characters in virtual world, we first investigate how a self-organizing neural model, known as TD-FALCON [14], may be exploited as autonomous learning agents in virtual world. TD-FALCON integrates temporal difference methods [10] and self-organizing neural networks [1, 2] for reinforcement learning with delayed evaluative feedback. Recently applied to a first-person shooter game, namely Unreal Tournament 2004 [12], the TD-FALCON agent has been shown to be capable of adjusting its tactical strategies and choosing appropriate weapons during run time similar to a human player. We extended TD-FALCON in a non-trivial way to address the challenges of learning in virtual world, including the lack of explicit goals and well defined evaluative feedback.

More importantly, to enable more dynamic and personalized experience, we propose to incorporate adaptive user modeling with our learning personal agents. Specifically, a two-channel fusion ART [1, 2], is used for learning player models during the players' interaction. By formulating cognitive codes associating agent's recommendation to user feedback, fusion ART learns direct player models of the users' likes and dislikes.

Building upon these learning and adaptive virtual agents, we aim to propose a cognitive agent model for virtual human. Three key requirements of realistic virtual humans in the virtual world have been identified, namely (1) autonomy: agents can function on their own; (2) interactivity: agents can interact naturally with players; and (3) personality: agents can exhibit human traits and characteristics. We present an integrated agent architecture extending from a family of self-organizing neural models known as Adaptive Resonance Theory (ART) [2, 15]. The agent model makes decisions not only based on the situational factors of what are perceived from the environment but also by taking into considerations of various properties, such as the context of both agents and players, desire, intention, and personality.

Figure 1.3 outlines the steps that were taken towards building human-like virtual agents in virtual worlds.

Figure 1.3: The flow of this dissertation.

1.4 Contributions

The aim of this dissertation is to simulate the “mind” part of virtual agents in order to make them more human-like. We formalize the three key requirements for developing human-like agents in virtual environment and present a framework for cognitive agents models. To this end, several tasks have been carried out. They are described as follows.

1.4.1 Reinforcement Learning in Virtual Worlds

As agents in a virtual world roam, interact and socialize with users and other agents as in real world without explicit goals and teachers, learning in virtual world presents many challenges not found in typical machine learning benchmarks.

We highlight the unique issues and challenges of building learning agents in virtual world using reinforcement learning and deployed a self-organizing neural model named, TD-FALCON to enable an autonomous agent to adapt and function in a dynamic environment with immediate as well as delayed evaluative feedback signals [TanKang10c]. We have implemented and evaluated TD-FALCON agents as virtual tour guides in a virtual environment. Our experimental results show that the agents are able to adapt and improve their performance in real time. The above work has been presented and included in the proceedings of IJCNN 2010 [KangTan10a] and Cyberworlds [Kang10d].

1.4.2 Learning Personal Agents with Adaptive Player Modeling

We aim to enable more dynamic and personalized experience by employing player modeling technology in virtual worlds. We propose to incorporate adaptive user modeling with our reinforcement learning agents. Specifically, a two-channel fusion ART [1, 2], is exploited for learning player models during the players’ interaction. By formulating cognitive codes associating agent’s recommendation to user feedback, fusion ART learns direct player models of the users’ like and dislike.

We have integrated learning personal agents with adaptive player modeling in a 3-D virtual world called Youth Olympic Village (YOV) Co-Space. Our experimental results show that we are able to learn player models that evolve and adapt with players during run time. Furthermore, the virtual tour guides with adaptive player models outperform those without player models. This work has been published in the proceedings of Intelligent Agent Technology (IAT) [KangTan10b].

1.4.3 Self-Organizing Cognitive Models for Virtual Agents

Working towards the challenges of building human-like characters, we propose a brain-inspired agent architecture that integrates goal-directed autonomy, natural language interaction and human-like personality. Extending from a family of self-organizing neural models known as Adaptive Resonance Theory (ART) [2, 15], an integrated agent architecture is proposed which makes decisions not only based on the situational factors of what are perceived from the environment but also by taking into considerations of belief, desires and personalities. In this way, these agent-based virtual humans are living a more realistic life by going to classes, having dinner with friends and chit-chatting with others. This work has been presented in the proceedings of Intelligent Agent Technology (IAT) [Kang12a].

1.4.4 Case Study and Evaluation of Agent based Virtual Human

As a showcase of our work, we have developed a Co-Space that models the Nanyang Technological University (NTU) campus. We have created realistic 3D models of almost all locations in NTU and populated the Co-Space with avatars that take on human appearance and body proportion. We have deployed agent based virtual humans into NTU Co-Space. The details of this work has been published in the proceedings of Autonomous Agents and Multiagent Systems (AAMAS 2012) [Kang12c].

In order to evaluate the effects of virtual humans on user experience, we have conducted a series of experiments with users where three instances of NTU Co-Space, each populated with a distinct type of virtual agents in the form of avatars, are assessed. In the first environment, agents could display static messages and do not have the capability

to interact with the users. In the second environment (E2), virtual humans are designed as embodied conversational agents using AIML [16] technology. The third environment (E3) is populated with intelligent autonomous virtual humans with our proposed fusion ART-based agent architecture. We captured seven constructs related to their user experience, namely *Telepresence*, *Social Presence*, *Perceived Interactivity*, *Perceived Usefulness*, *Flow*, *Enjoyment* and *Behavioral Intention* (to return to NTU Co-Space). The evaluative results show that the fusion ART-based agents deployed in E3 are more useful in assisting the subjects and E3 is significantly better in *Telepresence* and *Perceived Interactivity* than those of the other two environments. Moreover, the *Flow* experience perceived by the subjects in E3 (i.e., populated with the fusion ART-based virtual humans) is significantly stronger than that in E2 (with the AIML based virtual humans). The user study has been published in the proceedings of International Conference on Information Systems (ICIS 2012) [Kang12b] and Intelligent Virtual Agents (IVA 2013) [KangTan13].

1.5 Thesis Organization

The main body of the thesis is organized as follows.

Chapter 2 presents a literature review of the work related to developing human-like agents in virtual environments. It summarizes the existing technologies and puts forward the motivation of our work.

Chapter 3 highlights the three key requirements of realistic virtual humans in the virtual world and presents a framework for human-like cognitive agents.

Chapter 4 highlights the specific issues and challenges of building learning agents in virtual world. Specifically, a self-organizing neural model, named TD-FALCON (Temporal Difference - Fusion Architecture for Learning and Cognition), is presented, which enables an autonomous agent to adapt and function in a dynamic environment with immediate as well as delayed evaluative feedback signals.

Chapter 5 addresses a critical component for enabling personalized virtual world experience, which has the capability to adapt over time to the habits and eccentricity of a particular player. This chapter presents a cognitive agent with learning player model capability for personalized recommendations in a virtual world environment.

Chapter 6 presents a brain-inspired agent architecture based on self-organizing neural networks that integrates goal-directed autonomy, natural language interaction and human-like personality.

Chapter 7 reports a case study based on the model presented in chapter 6. The evaluation study involving human subjects was conducted to assess how virtual characters may enhance user experience in the virtual world.

The final Chapter summarizes the research contributions and results achieved so far, and further discusses the future work.

Chapter 2

Literature Review

In this chapter, related work is discussed in terms of the following five perspectives, namely intelligent agents in virtual worlds, player modeling in virtual worlds and computer games, modelling realistic virtual humans, cognitive models and evaluation of humans-like agents.

2.1 Virtual Agents

Intelligent agents have been popularly used for improving the interactivity and playability of virtual worlds. The key approaches and techniques are reviewed as follows.

2.1.1 Rule-Based Systems

In both real-world and virtual world AI applications, rule-based AI systems may be the most widely used AI systems. It defines a set of rules about states of the virtual world in following form: *if* $\langle condition \rangle$ *Then* $\langle action \rangle$. If an NPC's current state fulfils the $\langle condition \rangle$ of a rule, then the corresponding $\langle action \rangle$ is taken. Rule-based systems have several advantages: first, when facing a particular problem domain, they tend to simulate the way that people think and reason with a set of facts and knowledge. Secondly, by encoding the knowledge into rules which could be coded in any order, it is relatively easy to program and manage. Therefore, it is quite flexible to code the system as well as modifying the system at a later time [17]. However, rule-based system requires large number of pre-encoded rules to achieve high performance. Hence, it increases the designer and experts' burden of coding. Moreover, within the fixed set of predefined rules,

only limited adaptation could be occur. When facing new instance, it knows nothing. Finally, these rules always contain experts' bias which may not fit for every player.

A number of agents are based on Rule-Based approaches. For example, in the Virtual Theater project, synthetic actors portray fictive characters and provide improvising behaviors. The agents are based on a scripted social-psychological model with the defined personality traits which rely on the values of moods and attitudes [18]. Agents in Meta-verse, built using Active Worlds, are capable of performing the tasks typically associated with human beings, such as taking tickets for rides and acting as shopkeepers. However, these agents are basically reactive agents which work in a hard-coded manner. Virtual psychotherapist ELIZA [19], designed to take care of the 'patients', is also achieved with rule-based, adeptly modeled small talk. A conversational virtual agent Max has been developed as a guide to the HNF computer museum, where he interacts with visitors and provides them with information daily [20]. However, the design remains rule-based.

2.1.2 Finite State Machine

A finite state machine (FSM) is a well known behavior model containing a finite number of states, actions and the transitions between them. As illustrated in Figure 2.1, each potential state is represented as a circle and the transition functions are indicated as arrows. In this case, there are four possible states S_i, S_1, S_2, S_3 and five transition functions t_1, t_2, t_3, t_4, t_5 . It starts with the initial state S_i . Once the t_1 transition function provides a stimulus, the state transfers to S_1 . The rest of this FSM can be reason in the same manner. FSM is quite useful in an NPCs reasoning process. The process could be divide into a set of internal states and transitions [21, 22]. In this way, when and how the agent should respond to its environment are defined. Although FSM have been around for a long time, they are still commonly used in modern games and virtual worlds. This is because they are relatively easy to understand, implement, and debug. The Virtual Reality Gorilla Exhibit, an immersive virtual environment, teaches users about gorilla behaviors and social interactions. They used a state machine to control behavior selection which lets several transitions be valid at one time and uses a priority scheme to select which one to make [23]. Asbahr developed a portable virtual world simulation framework, namely Beyond, in which the SimObjects exhibit

autonomous behavior because of encoding the sensory inputs and effector outputs with a FSM, such as fear, anger, desire and energetic [24]. To name one of the most pioneer works, Khullar et.al have implemented the automating attending behaviors of virtual human characters also by finite state machine [25]. However, FSM has similar issues as rule-based systems. States and transitions are established before players interacting with their environment. Although different internal states may have different rules that would lead to more adaptable behavior than rule-based systems, the predefined rules within each internal state would still lead to the limited adaptation ability.

Figure 2.1: Generic finite state machine diagram

2.1.3 Fuzzy Logic

Fuzzy logic presents problems in a way which is quite similar to the way humans solve them. Everything in fuzzy logic is represented as degree. Specifically, it provides a method to deduct a result based on the vague, ambiguous, inaccurate, or incomplete facts [21]. A typical fuzzy inferring process is shown in Figure 2.2. After fuzzification, it uses fuzzy rules of the form if $\langle X \text{ is } A \rangle$ then $\langle Y \text{ is } B \rangle$. X and Y represent linguistic variables representing features being measured, while A and B are fuzzy categories which define decision thresholds. Fuzzy logic can be applied to many approaches such as rule-based systems and FSM, and used in virtual worlds in a variety of ways, such as controlling NPCs, assessing current situation posed by players and classify the characteristics of players and NPCs. Improv, a system for scripting interactive actors

in virtual worlds, is an example of making decisions by using fuzzy logic [26]. Another example is the fuzzy logic based intelligent agent supported personalization for virtual learning environments proposed by Xu [27]. However, while fuzzy logic gives the freedom to reason in uncertain environments, given the set of predefined fuzzy rules, the ability of characters to adapt is still limited.

Figure 2.2: Overview of fuzzy logic process

2.1.4 Decision Trees

In view of the limitations of static agents, some researchers have adopted learning methods into intelligent agents in virtual world. Decision trees are hierarchical graphs that structure Boolean functions [28]. Based on several input variables, it is able to predict the value of a target variable. Usually in decision trees, the nodes represent the input attribute and leaves represent classifications. An example of decision tree is provided in Figure 2.3. In virtual worlds, a decision tree is learned from a training set of previously made decisions, and it can then be used to make future decisions. In Treasure Hunt world, for example, the player together with a decision tree based companion agent search for treasure, while decision trees provide interpretable rules that support runtime decision making [29]. Since decision trees allow characters to learn from examples, it allows more adaptation than the previous approaches. However, decision trees are more suitable for discrete input values than numeric or continuous values. Most importantly, learning in decision trees is usually non-incremental, all the examples are required to store previously while in virtual world, examples may not always provided. Last but not least, decision trees are not able to adapt to changing environment during game play.

Figure 2.3: An example of decision trees

2.1.5 Neural Networks

Artificial neural networks (ANN) is constituted by a number of neurons with interconnecting pathways [28]. Although it is biological metaphor for neural networks, we could consider it as a mathematical function approximator. The input to the ANN represents independent variables and the output represents the dependant variables. The ANN itself is a non-linear function that gives a set of output by giving a set of input. Like decision trees, many ANN are supervised learning which learns from examples. The network is trained by providing the correct actions to perform under different situations as training data. Hence, when encountering a similar situation, the network are able to make a decision on the action to select based on the trained neural network. Being good at handling highly nonlinear problems, neural networks is highly efficient in simplifying complex state machines or rules-based systems to one or more trained ANN. In *Battlecruiser: 3000AD*, neural network has been used to guide NPCs to negotiate, trade and combat with players. However, in the case of learning from interaction, traditional neural networks are not adequate. This is because in interactive problems, it is often impractical to obtain the training data that are both correct and representative of all the situations. Furthermore, the network itself is like a black box, it is difficult to understand exactly the logic inside and how it is arriving at its results. Second, it's difficult to predict the generated output, especially if the network is programmed to learn or adapt within a gameplay. These make the testing and debugging of a neural network more difficult than previous approaches.

Figure 2.4: An example of artificial neural network

2.1.6 Evolutionary Agents

Evolutionary approaches simulate the process of biological evolution. It usually implements concepts like natural selection, reproduction, and mutation. Following the concept, individuals are defined in terms of a digital chromosome. When individuals are given birth, via the processes of crossover and mutation, offspring are defined by a combination of their parents chromosomes. The individual fitness of new individuals are evaluated to determine which will remain in the population and which will be removed. A four step process of genetic algorithms for games is illustrated in Figure 2.5. They basically involve a trial-and-error approach. And the solutions won't always be the same for every player. That's the beauty of genetic algorithms [17]. It will adapt to individual players. In second life, Novamente virtual agents have genetic-algorithm-like methods operating inside them, but these are combined with other methods in a sophisticated overall design [30]. However, when genetic algorithms are employed into creating NPCs, fitness functions must be predefined by designers as well. In this way, it contains experts' bias which may not always be suitable for the players and hence, limits the adaptability of a given population of individuals.

2.1.7 Bayesian Learning

Bayesian learning is based on a mathematical theory known as Bayes Theorem which allows us to determine a more realistic and probable assumption about the occurrence of

Figure 2.5: Overview of Evolutionary process

A if we know that B has occurred. Specifically, given the input data, Bayesian learning calculates the probability of each hypothesis or example and makes predictions on that basis. In other words, the predictions are made by using all the hypotheses (examples), weighted by their probabilities, rather than by using the “best” out of all examples. In this way, learning is reduced to probabilistic inference [28]. Because of these, the prediction confidence of Bayesian learning is much higher than some of the previous discussed approaches. In addition, it handles noise well. Bayesian learning has been proven to be extraordinarily helpful in getting the AI to reason about situations in a human-like fashion in games and virtual worlds [31]. Kopp et.al have made use of Bayesian learning to built agent that can decide if the production of a gesture is adequate, which representation technique to perform and which handshape to use given the evidence about the communicative context at hand [32]. Despite its numerous merits, it is computationally expensive and more importantly, it is less effective with low number of examples which may be commonly happened in virtual world.

2.1.8 Other Approaches

Other than the previous approaches, Yoon et.al. present a Creature Kernel framework to build interactive synthetic characters in the project Sydney K9.0 [33]. Their agents can reflect the characters’ past experience and allow individual personalization. But all the capabilities of the agents rely on past knowledge and couldn’t adapt to user gradually during run time. The co-present agents in a virtual gallery [34] utilize a knowledge base containing general input response knowledge, augmented with knowledge modules for special domains. More recently, an embodied conversational agent that serves as a virtual tour guide in Second Life has been implemented by Jan [5]. It uses NPCEditor [35] to learn the best output for any input from a training set of linked questions and answers.

Again, it learns from past experience but does not adapt over time according to the habits of a particular player or the changes in the environment.

All the work described above have developed a wide range of agents in virtual world with specific motivations. However, to the best of our knowledge, there has been very few in-depth work, if any, building complete agents that perform reinforcement learning and adapt their actions and behaviors in real time. Our work is motivated by such a consideration.

2.2 Player Modeling

The beneficial outcomes of player modeling in computer games and virtual worlds have recently attracted the interest in both the academia and the computer games industry.

2.2.1 Interactive Storytelling

Other than creating intelligent agents, in the domain of interactive storytelling, many researchers employed player modeling to learn the player's preferred style of play, and then uses that model to select the content of an interactive story [36, 37]. Similarly, Charles and McNeill have argued that player modeling may be used alongside existing approaches to facilitate player-centered game design, in the form of providing a more appropriate level of challenge, smoothening the learning curve, and enhancing the gameplay experience [38].

2.2.2 Poker Games

Comparing with most other games, poker games have been seen as a pioneer of player modeling in that, it already had a greater impact on success. Lockett and Miikkulainen made use of coarse approximations to game-theoretic player representations to improve the performance of software players in Limit Texas Hold 'Em poker [39]. Also, Baker et.al. investigated the impact of Bayesian opponent modeling upon the evolution of a player for a simplified poker game [40]. Unlike virtual world, player models are relatively simple in poker games. They are simply divided into three or four categories. In contrast, virtual world presents a much more complex environment. As such, simple strategies may not work.

2.2.3 Other game genre

Houlette defined the concept of player models and discussed the strategies for design, implementation and integration for general games such as First Person Shooting games [41]. However, he didn't present any experimental result to support his theory. Other researchers, such as Hladky and Bulitko, successfully predicted opponent positions by evaluating hidden semi-Markov models and particle filters in First Person Shooting games [42]. Heijden et.al. discussed an approach to organizing units by learning the effectiveness of a formation and directly applying learned formations according to the classification of the opponent player in Real-Time Strategy games [43]. However, all these work employ offline learning which can not deal with the changes in the players' behavior during run-time.

All the work described above have involved the use of player models in games with specific motivations. However, to the best of our knowledge, there has been very few work, if any, integrating player models with reinforcement learning agents in virtual world that can adapt to the habits and eccentric of a particular player dynamically. Our work is motivated by such a consideration.

2.3 Modeling Virtual Human

Over the past 15 years, a number of research groups have explored different aspects in modelling realistic virtual humans, including autonomous behaviors, interactive non-verbal and verbal behaviors and agents with personality and mood. Table 2.1 provides a summary of some representative systems in view of the three criteria, namely autonomy, interactivity and personality.

Different branches of works have been conducted based on various considerations. With the objective of implementing autonomous virtual humans, Merrick and Maher [46] created motivated reinforcement learning agent based virtual villagers in Second Life (SL). These agents explore their environment and learn new behaviors in response to interesting experiences, so that they could adapt and evolve over time. In Edward's work [48], agent based virtual workers simulate human's cognitive processes in operating in dangerous situations at a virtual site named SEVESO.

Table 2.1: Comparison of the representative virtual humans.

Name	Affiliation	Application	Auto-nomy	Inter-activity	Person-ality
Laura [44]	MIT	Real-Estate Agent	No	Yes	No
Max [20]	Bielefeld U.	Museum guide	No	Yes	No
Valerie [45]	CMU	RoboCceptionist	No	Yes	Yes
MRL [46]	U. of Sydney	Villagers	Yes	No	No
Justina [47]	USC	Virtual Patient	No	Yes	No
MASVERP [48]	UTC	Virtual operators	Yes	No	No
Eva [49]	U. of Geneva	Geography teacher	No	Yes	Yes

Another important branch of works is building interactive virtual humans in terms of verbal behaviors and non-verbal behaviors. Laura [44], initially designed to investigate human-computer relationships, is capable of carrying out conversations with players based on scripts and simple mechanisms for remembering past events with players. Max [20] is a conversational virtual agent which has been developed as a guide to the HNF museum. By applying the goal-driven BDI architecture, he is able to interact with visitors and provide them with information. Kenny et al. [47] developed a virtual patient named Justina who is able to act and carry on a dialogue like a real patient with a specific mental condition of interest.

In view of the limitations of monotonous interaction between virtual humans and real humans, many researchers have started to incorporate personality into virtual humans to make them more human-like. Valerie [45] was designed to help the visitors in many ways such as giving directions and checking weather forecast. Her facial expressions would change according to her mood which varies at different times. This creates a vivid feeling when visitors interact with her. Eva [49] was designed as a tutor of digital photography. It was developed based on a memory-based emotion model that used the memory of past interactions to build long-term relationships between the virtual character “Eva” and players.

The works described above have developed a wide range of virtual agents with different features. Our work reported in this paper is one of the few that have developed virtual human agents that exhibit autonomy, interactivity and personifications in real time to make virtual human agents as realistic as possible.

2.4 Cognitive Models

Research on integrated cognitive architectures is interdisciplinary by nature, spanning the fields of artificial intelligence, cognitive psychology, and neurobiology. Over the past decades, many cognitive architectures have been proposed and steadily developed, based on different approaches and methodologies [50].

2.4.1 State, Operator, and Result (Soar)

Soar is one of the first cognitive architectures proposed [51]. It incorporates a wide range of problem solving methods and learns all aspects of the tasks to perform them [52]. It is used for the understanding of the mechanisms required for intelligent behaviour and the incorporation of the mechanisms to form a general cognitive architecture in classical artificial intelligence.

Figure 2.6 illustrates the Soar architecture, consisting of the various memory structures and a decision making mechanism. The memory structures include long term memory and working memory. Information from the environment is made available in the working memory via perception, allowing appropriate actions to be chosen during domain-independent problem solving. Knowledge is stored in the long term memory, which can be classified into procedural, semantic and episodic memories. Procedural memory provides the knowledge of performing tasks. Semantic memory stores general facts about the world and is considered as declarative knowledge. On the other hand, episodic memory contains specific memory of an event experienced. The working memory in Soar cognitive architecture houses all the knowledge that are relevant to the current situation. The states (and sub-states) give the information on the current situation. The operator provides the steps to apply during problem solving.

2.4.2 Adaptive Control of Thought-rational (ACT-R)

ACT-R [53] provides a step by step simulation of human behaviour for detailed understanding of human cognition. In recent projects, ACT-R is also used in the prediction of activation patterns in brain with the aid of functional Magnetic Resonance Imaging (fMRI).

Figure 2.6: The Soar Cognitive Architecture.

In this cognitive architecture, information from the external environment and knowledge stored in the memories work conjunctively to select actions for execution to satisfy the goal(s) of the agent. The four basic modules in the ACT-R architecture are the visual, manual, declarative memory, and goal modules as illustrated in Fig. 2.7. The visual module is needed for identifying objects in the visual field and can be further classified into the visual location and visual-object modules. The visual-location module is required for determining the locations of objects, while the visual-object module aids in the recognition of the objects. The manual module is used for the control of the hand actuators. Information from memories can be retrieved via the declarative memory module, while the goal module keeps track of the agents current goals, and enables the maintenance of the agents thought in the absence of supporting external stimuli [53].

2.4.3 ICARUS

ICARUS [54] aims to unify reactive execution with problem solving, combine symbolic structures with numeric utilities, and learn structures and utilities in a cumulative manner. Different from other cognitive architecture, the design for ICARUS has been guided by the following principles: (1) Cognitive reality of physical objects; (2) Cognitive separation of categories and skills; (3) Primacy of categorization and skill execution; (4)

Figure 2.7: The ACT-R 5.0 model.

Hierarchical organization of long term memory; (5) Correspondence of long term or short term structure; and (6) Modulation of symbolic structures with utility functions.

Figure 2.8 presents an overview of the ICARUS architecture, consisting of four main components, namely the perceptual buffer, the conceptual memory, the skill memory, and the motor buffer. The perceptual buffer is involved in the temporary storage of percepts. Conceptual memory can be further classified into short term conceptual memory and long term conceptual memory. The skill memory is subdivided into short term skill memory and long term skill memory. The skill signals in the motor buffer, read out from the short term skill memory, are then executed by the ICARUS agent to bring about changes to its environment.

Figure 2.8: The schematic diagram of ICARUS.

2.4.4 Belief-desire-intention (BDI)

BDI is the theory of human practical reasoning [55]. BDI is designed to be situated, goal directed, reactive, and social which enables the agent to react to changes and communicate in their embedded environment, as it attempts to achieve its goals. BDI agents are typically implemented as Procedural Reasoning System (PRS).

Figure 2.9 shows the schematic of the BDI architecture, consisting of the main components of beliefs, desires, and intention, together with an interpreter. Beliefs are facts about the world as well as inference rules that may lead to acquisition of new beliefs. The beliefs are updated by the perception of the environment and the execution of the intentions [56]. Desires, also known as goals, are the objectives that a BDI agent aims to accomplish. Intentions are the actions that the agent is committed to perform in order to achieve the desires. Thus, it contains all the tasks chosen by the system for execution.

Figure 2.9: The BDI architecture in dMARS specification.

2.4.5 The Subsumption Architecture

In view that classical artificial intelligence often has the problems of extensibility, robustness, and achieving multiple goals, The subsumption architecture was proposed as an incremental and bottom-up approach to deal with these problems [57]. It decomposes a problem in terms of the behaviours exhibited by the robots instead of the stages of information flowing within the controller as in a traditional AI design.

Figure 2.10 depicts the subsumption architecture, comprising of a hierarchy of simple behaviour- based modules organized into layers (levels of competence). Subsumption allows all layers to access the sensors data and multiple (simple) behaviours to operate in parallel. Each layer is capable of controlling the system by itself, unlike classical artificial intelligent agents. Each level of competence displays a behaviour to pursue a particular goal and a higher level layer tends to subsume the underlying layers. The lower layers work like fast-adapting mechanisms, allowing the agent to react quickly to changes in its environment. In contrast, the higher layers control the system towards the overall goals. Consequently, the lower level behaviours are the results of the reactions towards

the environment, while the higher level behaviours are driven by the aim to pursue the primary goals. In order to achieve a higher level of competence, a new layer of control can be simply added without altering existing layers [57–59]. The layered design of the architecture is thus analogous to a biological nervous system, wherein new sections of brain are developed for new functions, but old sections are still preserved to perform their original tasks [60, 61].

Figure 2.10: The subsumption architecture.

2.4.6 Connectionist Learning with Adaptive Rule Induction Online (CLARION)

CLARION is a hybrid architecture that incorporates both implicit and explicit memories for reasoning and learning [62]. It is able to react in a dynamically changing environment without any preexisting knowledge installed into the architecture [63, 64].

Figure 2.11: The CLARION architecture.

As shown in Fig. 2.11, the CLARION architecture consists of two levels: a top level containing propositional rules and a bottom level (reactive level) containing procedural knowledge. The top level comprises of explicit symbolic mechanisms and the bottom level uses subsymbolic neural mechanisms. The architecture consults both the rules and the Q-values computed, combining their recommendations in a weighted sum, in order to select the most appropriate action to perform at each step. It is generally accepted that declarative knowledge is explicit, while procedural knowledge is implicit. As a result, declarative knowledge is usually accessible but procedural knowledge is not. However, there may be exceptions. Thus, the CLARION architecture includes non-action-centered subsystem (NACS) and action-centered subsystem (ACS). NACS contains mostly declarative knowledge, whereas ACS contains mainly procedural knowledge. The top level of NACS is a general knowledge store (GKS) and contains explicit representation. On the other hand, an associative memory networks (AMN) forms the bottom level of NACS,

which consists of implicit representation. In ACS, explicit action rules are stored in the top level. These rules are either provided by the agents external environment or extracted from information in the bottom level. The bottom level of ACS comprises of implicit decision networks, which can be trained by reinforcement learning [65, 66].

2.5 Evaluation of Human-like Agents

Although many approaches have been proposed for building human-like agents in virtual worlds, there have been relatively few studies in investigating the effect of virtual humans on user experience. We briefly review relevant work below.

Within Second Life, in the ancient City of Uruk 3000 B.C., virtual agent based non-player characters (NPCs) serve as the knowledge carriers as well as an important element in establishing the connection between the environment, objects and knowledge [67]. In social simulations, studies have shown that synthetic characters have provided an engaging and believable experience for the trainees in training leadership and negotiation tactics as well as in creating more dramatically interesting game situations [68]. Besides social simulations, virtual patients for the assessment and training of novice clinicians have been evaluated as well. The evaluation shows that they can provide valid, reliable and applicable representations of live patients [69]. Further evaluation also indicates that the interactions between novice clinicians and the virtual patients resulted in a compatible dialectic in terms of rapport, discussion of the traumatic event, and the experience of intrusive recollections [47].

Beyond the computer science field, some work has also been done in the domain of information systems. Chaturvedi et al. [70] have examined the design, development, validation, and use of virtual worlds, especially those using agent-based simulation as the underlying technology. They derive a set of design principles based on deep versus emergent structures while developing extended design principles to facilitate their integration. Nowak and Biocca [71] conducted an experiment to examine the influence of agents on presence, copresence, and social presence in a virtual environment. The result shows that people respond socially to both humans and agents. Barfield and Hendrix [72] found that agents shaking continuously or appearing to move between large discrete positions will likewise diminish a participant's sense of presence.

2.6 Summary

In this chapter, we have reviewed the existing approaches in five research areas, namely intelligent agent in virtual worlds, player modeling in games, virtual humans in virtual world, cognitive agent architecture and the evaluation of human-like agents. We discuss the inadequacy of existing methods and bring forward our motivation.

Chapter 3

Towards Building Human-Like Virtual Agents

3.1 Introduction

Research on building virtual humans has been extensive and ongoing for at least the past 15 years [5, 9, 13, 44, 73]. While early works focused more on the appearance and animation of the avatars, recent efforts are made mostly on making them more human-like. To achieve this ambitious goal, virtual humans need to overcome a number of challenges. First of all, they should be able to perceive and interact with the 3-D dynamics environment they reside. They should understand the context of the environment and other virtual characters and human users co-existing in the environment. Also, the reasoning process for choosing the appropriate action should preferably not be scripted or predetermined. Instead, they should function autonomously and make decisions proactively in response to events happening in the virtual world around them. Secondly, the interaction between virtual humans and real humans should be as natural as possible, integrating both verbal and nonverbal communication such as facial expressions and body gestures. The last but not the least, virtual humans should exhibit human-like traits and characteristics, such as personalities and emotions. All of these require the convergence of a diverse range of technologies, including computer graphics, 3D animation, cognitive science, natural language processing, decision making, dialogue management, speech and gesture generation and human-computer interaction [74]. More importantly, how to integrate these parts is a complex issue [68]. Working towards these challenges, in this chapter, we present the formulation of the problem and conceptual design.

3.2 Properties of Virtual Human

Building intelligent and human-like virtual characters are quite different from building traditional agent in many ways. It requires a convergence of several fields such as computer graphics, computer animation, artificial intelligence, human-computer interaction and cognitive science. On the other hand, it also has a close relationship to the area of robotics since they can share the same know-how in order to model the cognitive behaviour of autonomous individuals [74]. Despite the various applications using these virtual characters, for example, military training, remote education and entertainment, they all converge to the same set of properties, notably autonomous behaviour, adaptation to environmental constraints, understand user's intention, user modeling, natural language communication, personality and emotional modeling. For convenience, we group these properties into three general properties, namely autonomy, interactivity and personification. We discuss each of the properties in details.

3.2.1 Autonomy

As described in [74], autonomy refers to the ability of an agent to self-govern its behaviour and to perform actions without receiving explicit instructions from its developer. In any real-time application of agents, autonomy is always considered as one of the most important agent properties. Especially, in virtual worlds, fidelity of the virtual life to the real one is important and autonomy means less work for the designer, since it is a long and difficult process to design every situation exactly.

In a more general sense, many definitions on autonomy exist. According to Franklin [75], an autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future. Following that, eight sub-features for autonomy can be enumerated, namely situation awareness, independence, self-organized, emergent, adaptive, pro-active, pre-defined knowledge, and maintenance of belief.

- Situation Awareness: The agent is able to understand the situation and context of its environment, including other agents and human players.

- Independence: The agent is able to operate independently without the direct intervention of humans or others in order to fulfil its objectives. In other words, no central controller for directing and coordinating individuals exists;
- Self-organized: The agent is able to organize itself to achieve the complex behaviors.
- Emergent: The agent is able to exhibit reasoning process, not scripted or predefined. Moreover, the exhibited actions are reasonable.
- Adaptive: The agent is able to adapt its behaviors in response to changes in the environment where it is situated;
- Pro-activeness: The agent does not simply act in response to their environment, it is able to exhibit goal-directed behaviour by taking the initiative.
- Pre-defined knowledge: The agent is able to take in explicit knowledge when available so that it has some baseline to follow.
- Maintenance of belief: The agent is able to maintain internal representation of beliefs about itself and the external world.

3.2.2 Interactivity

Interactivity refers to the ability of an agent enabling it to interact and communicate with its surroundings, including other agents and human players. At the most basic level, the agent should be capable of producing a response to a stimulus, an event, a user's input, or a change in its environment. For information exchange with an user, this involves processing and understanding the user's input in order to generate an appropriate response within the context and situation [76].

A realistic virtual character should interact with human players just like a real human. To achieve this, an assembly of several components such as conversational abilities as well as non-verbal behavior (facial expressions, body gestures) are required. Natural language dialogue is a very important part of the interaction between human and machine. A believable dialogue system requires the integration of the functionalities such as natural language understanding and natural language generation, dialog management,

error correction and so on. Furthermore, non-verbal behaviors such as facial expression and body language would enhance the human-like aspect.

- Natural Language Processing: Agent should understand the natural language and able to generate the corresponding sentences.
- Dialog Management: A dialogue has an opening, body and closing and although the user takes control in most parts of the dialogue, the overall dialogue should be the result of a mixed initiative between user and agent.
- Non-verbal expression: Interactivity could be shown through non-verbal behaviors such as facial expression and body gestures as well.

3.2.3 Personification

Personification means giving human-like attributes to non-human objects and this issue is becoming important in the field of virtual humans in the recent years. While human-like attributes, such as emotion, may not be meaningful for traditional expert systems or decision support systems, agents in virtual world applications, such as games, storytelling systems, interactive dramas, training systems and therapy systems, are preferred to be as human-like as possible. In particular, social behavior of computer characters with emotion and personality increases the realism and quality of interaction. They also can replace real people in many service areas as museum guides or receptionists.

3.3 Preliminaries

A Markov Decision Process (MDP) provides a simple, precise, general, and relatively neutral way of talking about a learning agent interacting with its environment to achieve a goal. As such, it provides a bridge to biological efforts for understanding the mind. Incompletely known MDP was recognized as a natural problem formulation for reinforcement learning. Hence, before presenting the agent model, this section provides a summary of the MDP formulation of a single agent model in an unconstrained, stationary, fully-observable setting. Specifically, the behaviour model of a reinforcement learning agent A_g can be defined as a 4-tuple $\langle \mathcal{S}, \mathcal{A}, p, r \rangle$, where

- $\mathcal{S}$ is the set of states that the agent $\mathcal{A}_g$ may be in,
- $\mathcal{A}$ is the set of actions the agent $\mathcal{A}_g$ can perform,
- $p: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ defines the probabilistic state transition function,
- $r: \mathcal{S} \times \mathcal{A} \mapsto R$ defines the reward function.

As shown above, the probability that the agent goes to state $s' \in \mathcal{S}$ upon execution of action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ is denoted by $p(s' \mid s, a)$. We assume that, for any action, the corresponding transition matrix is stochastic: $\sum_{s'} p(s' \mid s, a) = 1, \forall s \in \mathcal{S}, a \in \mathcal{A}$. The agent obtains a reward of $r(s, a)$ if it executes an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$.

In a discrete-time fully-observable MDP, at each time step, the agent observes the current state of the system and chooses an action according to its *policy*. A policy is said to be *Markovian* (or *history-independent*) if the choice of action does not depend on the current state and time. If, in addition to that, the policy does not depend on time, it is called *stationary*. A *deterministic* policy always prescribes the execution of the same action in a state, while a *randomized* policy chooses actions according to a probability distribution.

In an unconstrained discounted MDP, the learning goal of the agent is to find a policy that maximizes the total expected discounted reward over an infinite time horizon:

$$U_\gamma(\pi, \alpha) = \mathbf{E}\left[\sum_{t=0}^{\infty} (\gamma)^t r_t(\pi, \alpha)\right]. \quad (3.1)$$

where $\gamma \in [0, 1]$ is the discount factor, r_t is the reward the agent receives at time t , whose distribution depends on the policy π and the initial distribution over the state space $\alpha: \mathcal{S} \mapsto [0, 1]$

One of the most important results in the theory of MDPs states that, for an unconstrained discounted MDP with the total expected reward optimization criterion, there always exists an optimal policy that is stationary, deterministic, and uniformly optimal, where the latter term means that the policy is optimal for all distributions over the starting state.

3.4 A Framework for Human-Like Agent Models

In this section, we provide a general and formal definition of the human-like agent model by extending the traditional MDP for reinforcement learning to incorporate the three

human-like agent properties. Comparing with the existing models, the cognitive agent model is designed to provide a unified framework for integrating the three key properties of agents as virtual human. Specifically, the agent model incorporates explicit notions of Belief, Desire, and Personality, which conjunctively drive the agents' reasoning and decision making process.

As shown in Figure 3.1, the cognitive agent model consists of a *Perception Module* for receiving situational signals from the environment through a set of sensory APIs and an *Action Module* for performing actions through the various actuators. The *Inference Engine* then identifies the most appropriate action, by tapping into the belief, in accordance to the desire and personality of the virtual agent. The belief learned and used by the Inference Engine includes declarative knowledge of self, players, and environment, as well as procedural knowledge of goal-oriented rules, which guide an agent in fulfilling goals, and social rules, for generating socially appropriate behavior.

The detailed definition of the model is formally provided in the following sections.

Figure 3.1: A schematic diagram of the cognitive agent model.

3.4.1 Sensory-Motor Layer

The sensory-motor layer of an agent constitutes the sensory and motor functions through which the agent interacts with the environment that it is situated in. The formal definitions of the perception and action are given below.

DEFINITION 1 (Perception): Let $\mathcal{S}$ be the set of situational attributes perceived by the agent from its environment.

DEFINITION 2 (Actions): Let $\mathcal{A}$ be the set of commands for controlling the actuators of the agent.

3.4.2 Mental Layer

The mental layer of an agent consists of the long-term memory as well as the short-term mental states of the agent, serving as the basis for cognitive functions, such as reasoning and decision making. While the short-term memory maintains the active mental state of the agent, the long-time memory contains the agent's desire and personal characteristics. They are defined formally as follows.

DEFINITION 3 (Belief): The belief of an agent is a three-tuple $\mathcal{B} = \langle \mathcal{B}_s, \mathcal{B}_e, \mathcal{B}_u \rangle$, where $\mathcal{B}_s$ contains the knowledge of the agent about itself, $\mathcal{B}_e$ consists of the knowledge of the agent about the environment, and $\mathcal{B}_u$ represents the knowledge of the agent about the players and other agents in the same environment.

The belief of the agent takes part in the reasoning function of the agent. New belief can be generated as a result of a reasoning process. The belief of an agent may also be revised through a knowledge updating function or learning, through the agent's interaction with its environment and other agents.

DEFINITION 4 (Desire): The desire of an agent $\mathcal{A}g$ is a set $D = \{d_1, \dots, d_n\}$, where d_j is a target state which the agent aims to achieve.

At a given time t , any of the target states can be adopted as a goal for active pursuit by the agent. The desire D thus serves to guide the agent's behavior towards fulfilling

the goals. The agent is always intentional and continually monitoring its internal mental conditions and may update its desire if necessary.

DEFINITION 5 (Personality): The personality of an agent $\mathcal{A}g$ is defined as a set $\mathcal{P} = \{p_1, \dots, p_n\}$, where p_j indicates a personal attribute of the agent.

The personality of an agent adds biases and uniquely influences its cognitive processes, notably reasoning and decision making, in coping with various situations.

3.4.3 Cognitive Agent Model Formulation

The cognitive model for an agent $\mathcal{A}_{ca}$ is defined as a six-tuple $\langle \mathcal{S}, \mathcal{A}, r, \mathcal{B}, \mathcal{D}, \mathcal{P} \rangle$, where

- $\mathcal{S}$ is the set of perceptual inputs received by the agent,
- $\mathcal{A}$ is the set of actions the agent can execute,
- r is the reward function of the agent,
- $\mathcal{B}$ is the set of belief of the agent,
- $\mathcal{D}$ is the set of desire of the agent,
- and $\mathcal{P}$ is the set of personal characteristics of the agent.

The reward function serves the similar purpose of shaping the agent's behaviour following its actions to the environment. However, in the cognitive agent model, the reward function to be learned is a combinative outcome of the agent's state, action, desire and personality.

DEFINITION (Reward Function): The reward function r is defined by $r: \mathcal{S} \times \mathcal{A} \times \mathcal{D} \times \mathcal{P} \mapsto R$ such that the agent obtains a reward of r when it executes an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$, with a personality $v \in \mathcal{P}$ and desire $d \in \mathcal{D}$.

Different from the traditional reinforcement learning agent, the inference engine of the cognitive agent consists of two main functions, namely reasoning and decision making, as defined below.

DEFINITION 6 (Reasoning): The reasoning process of the agent is a function $\mathcal{R}_f: \mathcal{S} \times \mathcal{B} \mapsto \mathcal{S}$, where new belief is generated and added into the new state $s' \in \mathcal{S}$, based on the current state $s \in \mathcal{S}$ and the belief $\mathcal{B}$ of the agent.

DEFINITION 7 (Decision Making): The decision making process of the agent is a function $\mathcal{D}_f : \mathcal{S} \times \mathcal{D} \times \mathcal{P} \mapsto \mathcal{A}$, where the agent selects an action $a \in \mathcal{A}$ based on the current state $s \in \mathcal{S}$, in accordance to its personality $p \in \mathcal{P}$ and desire $d \in \mathcal{D}$.

Based on the current state of its perception, belief, desire and personality, the agent performs the functions of reasoning, decision making and learning iteratively until all its active goals are achieved.

Based on the above formulation, the extended-Markov Decision Process for the cognitive agent model can be defined as

$$\mathbf{M}_{ca} = (\mathcal{S}, \mathcal{A}, \mathcal{D}, \mathcal{P}, p(s', r|s, a, d, p)) \quad (3.2)$$

Consequently, the learning goal of the agent is to find a policy that maximizes the total expected discounted reward over an infinite time horizon:

$$\mathbf{U}_r(\pi) = \mathbf{E}[\sum_{t=0}^{\infty} (r)^t r_t(\pi, a)]. \quad (3.3)$$

3.5 Summary

For creating realistic virtual humans in the virtual world, three key features namely autonomy, interactivity and personification, are discussed in this chapter. Specifically, we have provided a problem formulation and presented a schematic agent model that integrates goal-directed autonomy, natural language interaction and human-like personality.

In the following chapters, we will first discuss the issues achieving autonomy, followed by the incorporation of user model to make virtual human more personalized. Finally, implementation of the integrated cognitive agent model is presented to meet the problem formulation discussed in this chapter.

Chapter 4

Reinforcement Learning in Virtual Worlds

4.1 Introduction

As agents in a virtual world roam, interact and socialize with users and other agents as in real world without explicit goals and teachers, learning in virtual world presents many challenges not found in typical machine learning benchmarks. In this chapter, we highlight the unique issues and challenges of building learning agents in virtual world using reinforcement learning, which include the presence of implicit goals, timing of learning, calculating of evaluative feedback and learning in a dynamic, real-time environment. Subsequently, a self-organizing neural model named, TD-FALCON is deployed, which enables an autonomous agent to adapt and function in a dynamic environment with immediate as well as delayed evaluative feedback signals. We have implemented and evaluated TD-FALCON agents as virtual tour guides in a virtual world environment. Our experimental results show that the agents are able to adapt and improve their performance in real time. To the best of our knowledge, this is one of the few in-depth works on building complete learning agents that adapt their behaviors through real time reinforcement learning in virtual world. This work has been published in [KangTan10a, KangTan10d].

4.2 Problem Formulation

We start with embedding intelligent agents in complex dynamic uncertain environments, as in virtual worlds. Following the Markov Decision Process (MDP), the environment can be modelled as a set of world state W . An agent typically partially observes the world and perceives a set of state $S = \{S_1, S_2, \dots, S_i\}$ which is a subset of W . After performing an action A upon the input S , there is some possibility that it will receive a feedback r generated by players. Meanwhile, the world state will transit accordingly and the above process is repeated. Hence, at each time t , the agent perceives its state $s_t \in S$ and the set of possible actions $A(s_t)$ and receives from the environment the new state s_{t+1} and a reward r_t . Based on these interactions, the reinforcement learning agent must develop a policy $\pi : S \rightarrow A$ which maximizes the quantity $R = r_0 + r_1 + \dots + r_n$ for MDPs. An illustration of reinforcement learning agents interacting with environments through sensors and actuators is shown in Figure 4.1.

Figure 4.1: An agent interacting with the environment

Based on the above formulation, it's obvious to observe that reinforcement learning is a good model for addressing this kind of problem because it offers a means of autonomous learning for agents that does not require direct supervision or examples by human players.

In reinforcement learning, rewards serve to define optimal policies in Markov decision processes (MDPs). An optimal policy is a policy that maximizes the expected total reward. The task of reinforcement learning is to use the observed rewards to learn an optimal policy for the environment. Furthermore, by incorporating temporal-difference (TD) method, it computes an estimated payoff value for the purpose of learning when the immediate evaluative feedback is absent.

4.3 TD-FALCON as Virtual Agents

TD-FALCON [14] incorporates Temporal Difference (TD) methods to estimate and learn value functions of action-state pairs $Q(s, a)$ that indicates the goodness for a learning system to take a certain action a in a given state s . Such value functions are then used in the action selection mechanism, also known as the *policy*, to select an action with the maximal payoff. Given the current states, TD-FALCON algorithm [14] selects an action with the maximal Q-value in the state by enumerating and evaluating each available action. The architecture of TD-FALCON is shown in Figure 4.2.

Figure 4.2: TD-FALCON architecture.

There are several considerations making TD-FALCON a suitable candidate for building learning agents in virtual world. We provide a discussion of the key enabling properties below.

4.3.1 Self-Adaptation

TD-FALCON learns through reward signals, which does not involve any human supervision and intervention. In this way, the agent does not need a perfect model to learn from. The agent is able to adjust its behaviors in different situations, if enough training is provided.

4.3.2 Generalization

Memory consumption is a major limiting factor to the size of the problems that a learning agent can tackle. Classical approaches of reinforcement learning usually causes a scalability issue for continuous and/or very large state and action spaces since policy function and/or value function must be learned for each and every possible state or possible pair of state and action. A self-organizing neural network can ease the memory requirement by its ability of generalizing and finding patterns in data. By incorporating self-organizing neural network, TD-FALCON can deal with the problem of learning a massive amount of data in virtual environment.

4.3.3 TD Learning

It is important to note that reward signals may not always be available in a virtual world environment. TD-FALCON has incorporated temporal difference methods to estimate and learn value function of state-actions pairs $Q(s, a)$ based on a combination of current and future estimated rewards. By making use of TD methods, they can perform multiple-step prediction in virtual world, in which the merit of an action can only be known after several steps into the future.

4.3.4 Fast and Stable Real-time Learning

Based on Adaptive Resonance Theory (ART), TD-FALCON is capable of performing fast and stable reinforcement learning in real time. Indeed, it has shown in a prior study that TD-FALCON produces a stable performance in a pace much faster than those of standard gradient-descent based reinforcement learning systems [14].

4.4 Learning Algorithm

4.4.1 FALCON Dynamic

FALCON employs a three-channel architecture (Figure 4.2), comprising a category field F_2 and three input fields, namely, a sensory field F_1^{c1} for representing current states, a motor field F_1^{c2} for representing actions, and a feedback field F_1^{c3} for representing reward values [77]. The generic network dynamics of FALCON, based on fuzzy ART operations [78], is described as follows.

Input vectors: Let $\mathbf{S} = (s_1, s_2, \dots, s_n)$ denote the state vector, where $s_i \in [0, 1]$ indicates the value of sensory input i . Let $\mathbf{A} = (a_1, a_2, \dots, a_m)$ denote the action vector, where $a_i \in [0, 1]$ indicates the preference of a possible action i . Let $\mathbf{R} = (r, \bar{r})$ denote the reward vector, where $r \in [0, 1]$ is the reward signal value and $\bar{r}$ is given by $\bar{r} = 1 - r$. The whole input vectors are with complement coding.

Activity vectors: Let $\mathbf{x}^{ck}$ denote the F_1^{ck} activity vector for $k = 1, \dots, 3$. Let $\mathbf{y}$ denote the F_2 activity vector.

Weight vectors: Let $\mathbf{w}_j^{ck}$ denote the weight vector associated with the j th node in F_2 for learning the input patterns in F_1^{ck} for $k = 1, \dots, 3$. Initially, F_2 contains only one *uncommitted* node. An *uncommitted* node is one which has not been used to encode any pattern and its weight vector contains all 1s.

Parameters: The FALCON's dynamics is determined by choice parameters $\alpha^{ck} \geq 0$, learning rate parameters $\beta^{ck} \in [0, 1]$, contribution parameters $\gamma^{ck} \in [0, 1]$ and vigilance parameters $\rho^{ck} \in [0, 1]$ for $k = 1, \dots, 3$.

The FALCON pattern processing cycle comprises five key stages, namely code activation, code competition, activity readout, template matching, and template learning, as described below.

Code activation: A bottom-up propagation process first takes place in which the activities of the category nodes in the F_2 field are computed. Specifically, given the activity vectors $\mathbf{x}^{c1}, \mathbf{x}^{c2}, \mathbf{x}^{c3}$, for each F_2 node j , the choice function T_j is computed as follows:

$$T_j = \sum_{k=1}^3 \gamma^{ck} \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_j^{ck}|}{\alpha^{ck} + |\mathbf{w}_j^{ck}|}, \quad (4.1)$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the norm $|\cdot|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code competition: A code competition process follows under which the F_2 node with the highest choice function value is identified. The system is said to make a choice when at most one F_2 node can become active after the code competition process. The winner is indexed at J where

$$T_J = \max\{T_j: \text{for all } F_2 \text{ node } j\}.$$

When a category choice is made at node J , $y_J = 1$ and $y_j = 0$ for all $j \neq J$. This indicates a winner-take-all strategy.

Activity readout: The chosen F_2 node J performs a readout of its weight vectors into the input fields F_1^{ck} such that

$$x^{ck(new)} = x^{ck(old)} \wedge w_J^{ck}. \quad (4.2)$$

The resultant F_1^{ck} activity vectors are thus the fuzzy AND of their original values and their corresponding weight vectors.

Template matching: Before the node J can be used for learning, a template matching process checks that the weight templates of node J are sufficiently close to their respective input patterns. Specifically, resonance occurs if for each channel k , the *match function* m_J^{ck} of the chosen node J meets its vigilance criterion ρ^{ck} :

$$m_J^{ck} = \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck}|}{|\mathbf{x}^{ck}|} \geq \rho^{ck}. \quad (4.3)$$

If any of the vigilance constraints is violated, mismatch reset occurs in which the value of the choice function T_J is set to 0 for the duration of the input presentation. The search process then selects another F_2 node J until a resonance is achieved.

Template learning: Once a resonance occurs, for each channel ck , the weight vector $\mathbf{w}_J^{ck}$ is modified by the following learning rule:

$$\mathbf{w}_J^{ck(new)} = (1 - \beta^{ck})\mathbf{w}_J^{ck(old)} + \beta^{ck}(\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck(old)}). \quad (4.4)$$

When an uncommitted node is selected for learning, it becomes *committed* and a new uncommitted node is added to the F_2 field. Fusion ART thus expands its network architecture dynamically in response to the input patterns.

4.4.2 TD-FALCON

TD-FALCON integrates temporal difference learning with the FALCON dynamics described above for reinforcement learning. The general sense-act-learn algorithm is summarized in Algorithm 1.

Algorithm 1: Generic Flow of The TD-FALCON Algorithm

- 1 Initialize the FALCON network;
 - 2 Given the current state s ;
 - 3 **while** s is not a terminal state **do**
 - 4 for each available action a in the action set $\mathcal{A}$, predict the value of the action $Q(s,a)$ by presenting the corresponding state and action vectors $\mathbf{S}$ and $\mathbf{A}$ to FALCON;
 - 5 Based on the value functions computed, select an action a from $\mathcal{A}$ following an action selection policy;
 - 6 Perform the action a , observe the next state s' , and receive a reward r (if any) from the environment;
 - 7 Estimate the value function $Q(s, a)$ following a temporal difference formula given by $\Delta Q(s, a) = \alpha TD_{err}$;
 - 8 Present the corresponding state, action, and reward (Q-value) vector, namely $\mathbf{S}$, $\mathbf{A}$, and $\mathbf{R}$, to FALCON for learning;
 - 9 Update the current state by $s = s'$;
-

As mentioned earlier, agents do not learn all the time in the virtual world. Therefore, TD-FALCON first needs to detect an opportunity for learning. The cues for learning typically include a change in the situation (state) and a receipt of an evaluative feedback. Given the current state s , the FALCON network is used to predict the value of performing each available action a in the action set $\mathcal{A}$ based on the corresponding state vector $\mathbf{S}$ and action vector $\mathbf{A}$. The value functions are then processed by an action selection strategy (also known as policy) to select an action.

After performing the action, a feedback (if any) from the environment is received. In the virtual worlds, the feedback could be a synthesis of the user's verbal expression, body language or emotional expression. Upon receiving the feedback, a TD-formula is used to compute a new estimate of the Q-value for performing the chosen action in the current state.

A typical Temporal Difference (TD) equation for iterative estimation of the value functions $Q(s,a)$ is given by

$$\Delta Q(s, a) = \alpha TD_{err}, \quad (4.5)$$

where $\alpha \in [0, 1]$ is the learning parameter and TD_{err} is a function of the current Q-value predicted by FALCON and the Q-value newly computed by the TD formula.

Using the Q-learning rule, the temporal error term is computed by

$$TD_{err} = r + \gamma \max_{a'} Q(s', a') - Q(s, a), \quad (4.6)$$

where r is the immediate reward value, $\gamma \in [0, 1]$ is the discount parameter, and $\max_{a'} Q(s', a')$ denotes the maximum estimated value of the next state s' .

This new Q-value is then used as the teaching signal (represented as reward vector $\mathbf{R}$) for FALCON to learn the association of the current state and the chosen action to the estimated value. The above steps repeat until a terminal state s is reached. This happens when the user has completed his/her journey of exploring the virtual world.

4.5 A Case Study on YOV Co-Space

Co-Spaces are virtual worlds developed for mirroring a real physical world in terms of look-and-feel, functions and services. The objective of Youth Olympic Village (YOV) Co-Space is to introduce the YOV and the hosting country to visitors around the world in an interactive and playable manner. To achieve this objective, we are in the process of developing and populating human-like cognitive agents in the form of autonomous avatars that roam in the landscape of YOV Co-Space. The agents are designed to be aware of its surrounding and can interact with users through their human avatars. With the autonomous avatars befriending and providing personalized context-aware services to human avatars, we aim to make the content and services readily available to the users.

The architecture of Co-Space is shown in Figure 4.3. As illustrated in this framework, the TD-FALCON based personal agent works in conjunction with the search agent in recommending functions and services to the users. Specifically, the personal agent determines the appropriate type of services to recommend whereas the search agent retrieves the specific services based on the environment situations as well as the users' context parameters. Figure 4.4 provides a screenshot of the virtual world, showing the personal agent Carol serving the user.

Figure 4.3: The architecture of Co-Space.

4.5.1 Embodiment of TD-FALCON in Co-Space

When applying TD-FALCON, challenges include representing the state and action space as well as computing the reward signals.

4.5.1.1 State Representation

The input state of the personal agent consists of five sets of attributes, namely time, location and the player's request, interest and current activity. A summary of these attributes together with the possible values is given in Table 4.1. All attributes adopt a binary encoding scheme in the state vector. Although we include user request as part of the state space, the agents are intended to work without explicit user request. In other words, they are supposed to proactively make recommendation based on user's interest, context and situation.

Figure 4.4: A screenshot of TD-FALCON based personal agent in Co-Space.

Table 4.1: Attributes and possible values in the state space

Attributes	Values
Time	Morning, lunch Time, Afternoon, Dinner Time, Evening, Bed Time
Player's location	Central, North, South, East, West
Player's request	Hotel, Dining, YOG Venue, Place of Interest(POI), Shopping, Unknown
Player's interest	Dining, YOG, POI, Shopping, Unknown
Player's activity	Hotel, Dining, YOG, POI, Shopping, Unknown

4.5.1.2 Action Space Representation

The personal agent is designed to determine the most appropriate service for recommendation to its user. The action field thus consists of recommendations of five types of services, namely hotel, dinning, YOG, place of interest and shopping. Using a binary encoding scheme, the action vector is represented as $\mathbf{a} = (a_1, a_2, \dots, a_5)$, where $a_j = 1$ and $a_k = 0$ for all $k \neq j$ indicate that action j is recommended.

4.5.1.3 Computing Reward Signals

The reward function $r(t)$ is defined as a synthesis of the player’s feedback and mood as $r(t) = (f(t) + m(t))/2$, where f is an explicit reward based on the user feedback through the dialog menu and m is an implicit reward value suggested by the gestures of the user. The dialog menu enables a player to choose among “wonderful”, “thanks”, “good”, “fair”, and “leave me alone”, which correspond to a reward value of 1, 0.75, 0.5, 0.25 and 0 respectively. Meanwhile, we gauge a player’s mood by observing the gestures of the player avatar. Gestures, such as waving, dancing, and jumping, indicate a positive reward of 1 since they indicate satisfaction with the service. In contrast, a reward of 0 is given to gestures, such as angry and walk away, which suggest unhappiness.

4.5.2 Empirical Experiments

We build a TD-FALCON network comprising 33 nodes in the sensory field, five nodes in the action field, and two nodes in the reward field. We conduct four sets of experiments to analyze how TD-FALCON performs in various settings, including learning in real time, learning with pre-inserted rules, adapting to changing user models, and learning with delayed feedback. In order to evaluate our system performance empirically, we design an automatic test procedure using synthetic user models. We simulate the user’s feedback to the agent’s service by using a set of user logic. By matching the agent’s recommendations with the user’s expectation, we are able to compute reward signals that can be used to guide the agent in learning. A sample set of user logic is given in Table 4.2 as an illustration. In the first example, when Time=Lunch or Dinner Time, Activity != Dinning, Request = Unknown, if the output of TD-FALCON is “recommend restaurant”, a reward of value 1 is given. Otherwise, a reward of value 0 is given.

4.5.2.1 Performance Measures

As the main role of personal agents is to provide services to the users, one important performance indicator is the accuracy of the service given to the users over a period of time. In addition, in the motivation of elevating the user experience in virtual world, we define a general goal of maximizing the user’s satisfaction. For a recency-biased indication of the system’s performance, we compute the user’s satisfaction index at time

Table 4.2: User logic for generating rewards.

IF	Time=Lunch or Dinner Time, Activity != Dinning, Request = Unknown,
EXPECT	Recommend restaurant
IF	Time=Lunch or Dinner Time, Activity = Dinning, Request = Unknown, Interest contains X
EXPECT	Recommend X
IF	Time = Bed Time, Request = Unknown
EXPECT	Recommend Hotel

t by $S(t) = \gamma r(t) + (1 - \gamma) S(t - 1)$, where $S(0) = 0$, $\gamma \in [0, 1]$ is the recency weighting factor, and $r(t)$ indicates the reward value received at time t .

4.5.2.2 Learning with User Feedback

TD-FALCON uses a default set of parameter values as listed in Table 4.3. The recommendation accuracy without pre-inserted rules and user satisfaction index ($\gamma=0.25$) averaged at 100-trial intervals over 3000 trials are shown in Figure 4.5 and Figure 4.6 respectively. The results, obtained after averaging across five sets of experiments, indicate that the agents are able to gradually improve their performance through user feedback continuously over time. We have also conducted experiments with different sets of vigilance parameter values. As summarized in Table 4.4, vigilance parameters have a considerable influence on the system: a higher vigilance produces highly detailed memories, while a lower vigilance results in more general memories. Nevertheless, we see that the performance of the agents are generally stable over a reasonable range of parameter values. Reducing the vigilance values drastically however causes a degrade in the system's performance.

4.5.2.3 Experimenting with Pre-existing Rules

In this set of the experiments, we investigate the performance of TD-FALCON with the pre-existing knowledge in the form of symbolic rules as illustrated in Table 4.5. A total

Table 4.3: TD-FALCON parameter setting.

FALCON Parameters	Values
Choice parameter($\alpha^{c1}, \alpha^{c2}, \alpha^{c3}$)	0.1, 0.1, 0.1
Learning rates($\beta^{c1}, \beta^{c2}, \beta^{c3}$)	1.0, 1.0, 1.0
Contribution parameter($\gamma^{c1}, \gamma^{c2}, \gamma^{c3}$)	1/3, 1/3, 1/3
Baseline vigilance parameter($\rho^{c1}, \rho^{c2}, \rho^{c3}$)	1.0, 1.0, 1.0
TD Learning Parameters	
TD learning rate α	1
Discount factor γ	0
Initial Q-value	0
ϵ-greedy Action Policy Parameters	
Initial ϵ value	0.5
ϵ decay rate	0.0002

Figure 4.5: Recommendation accuracy of TD-FALCON with and without pre-inserted rules.

of 180 rules are generated based on a selected user model. Each of these rules is inserted into TD-FALCON with a reward value of 1 before learning commences.

Comparing to the 33,480 possible combinations of state and actions, the 180 rules represent only a small fraction of the knowledge. Figure 4.5 and Figure 4.7 depict the com-

Figure 4.6: User satisfaction index obtained over time.

Table 4.4: Performance of TD-FALCON with different parameter value settings.

$\rho^{c1}, \rho^{c2}, \rho^{c3}$	Success rate at 3000 decision cycles	Number of Nodes
1.0 1.0 1.0	97%	523
1.0 1.0 0.5	99%	515
0.9 0.2 0.5	98%	518
0.8 0.2 1.0	78%	178

parative performance of TD-FALCON with pre-inserted rules and without pre-inserted rules in terms of recommendation accuracy and user satisfaction index respectively. We can observe that with the pre-inserted rules, the performance generally increases slightly faster in the first few hundreds of the interaction cycles comparing to that without pre-inserted rules. However, their performance are largely similar after 1000 interaction cycles.

Table 4.5: A sample set of pre-inserted rules.

IF	Time = Lunch Time or Dinner Time, Request = Unknown, Interest = POI and YOG, Activity = Hotel
THEN	Action = Recommend Dining
IF	Time = Bed Time, Request = Unknown, Interest = POI and YOG, Activity = Hotel
THEN	Action = Recommend Hotel
IF	Time = Morning or Afternoon or Evening, Request = Unknown, Interest = POI and YOG, Activity = Hotel;
THEN	Action = Recommend YOG or POI

Figure 4.7: User satisfaction index with and without pre-inserted rules.

4.5.2.4 Adapting to Changing User Logic

In this set of the experiments, we evolve a user model during run-time to investigate how the agents may cope with a shift in user's interests. We change 10% of the rules after 3000 learning trials to shift the user's interest to dinning and shopping. The expected action will be changed accordingly. For example, for case 2 in Table 4.2, the expectation will be "Recommend dinning or shopping" rather than the previous interest. As 10% of the user

logic for generating rewards has been changed, when the agent tries to use previously learnt knowledge to make a decision, it will definitely suffer a decline in accuracy.

To put the adaptation performance of TD-FALCON into perspective, we build an agent based on a set of rules, which readily achieves 100% accuracy at the beginning of the experiments. As shown in Figure 4.8 and Figure 4.9, the accuracy of the rule based system (RBS) suffers a significant drop and hovers around 60% after the change. This is because when the user model is changed, it does not have the capability of adapting to it. In contrast, TD-FALCON, despite suffering a drop to an accuracy of 55%, quickly adapts to the new user. After 1000 decision cycles, it manages to regain an accuracy of more than 90%.

Figure 4.8: Recommendation accuracy of TD-FALCON and the rule based agent (RBS).

4.5.2.5 Learning with Delayed Feedback

It is significant to note that all the above mentioned experiments rely on the feedback obtained after performing each action. However, in a realistic environment, it may take a long sequence of actions before a reward or penalty is finally given. This is the reason

Figure 4.9: User satisfaction index obtained by TD-FALCON and the rule based system (RBS).

why we incorporate TD method to estimate the credit of an action based on what it will lead to eventually rather than learning a function mapping states to actions directly.

In order to observe the effect of temporal difference learning, we evaluate TD-FALCON with a set of synthetic users, who may or may not provide feedback to each and every recommendations given by the agents. As a guide, the users respond to roughly 50% of the recommendations given. For the rest of the cases, the immediate reward value is zero and the agents rely on the Q-value computed using the TD-learning rule. The performance of TD-FALCON, in terms of prediction accuracy and user satisfaction index, can be found in Figure 4.10 and Figure 4.11 respectively.

We note that learning with delayed feedback is a much more difficult task due to the lack of explicit immediate reward signals and the large variety of the input instances. For both performance measures, we observe that the overall performance is degraded comparing with those obtained in the previous experiments. However, a reasonable level of performance is still maintained as TD-FALCON manages the conflicting requirement of short-term reactivity and long-term goal-driven behavior.

Figure 4.10: Average recommendation accuracy (with range bars) of TD-FALCON agents without immediate rewards.

4.5.3 User case study

We invited several people to be the trial users of YOY Co-Space. These people have different interest profiles and different personal attitudes towards giving feedback rewards to system (Active/ Passive). We inserted 300 rules initially and let users try out the system. These rules represent some common knowledge of human-being. The criterion for measuring the performance is the success rate of recommending services.

As shown in Table 4.6, although facing users with different interests and attitudes, our learning agent still achieved a good success rate of 83.6% on average. For people whose interest is relatively simple and active in providing feedback, the success rates are correspondingly higher than the others. As a matter of fact, this performance pattern is in accordance with that of a human tour guide as well.

Figure 4.11: User satisfaction index of TD-FALCON agents without immediate rewards.

Table 4.6: Result of real user study.

userID	Interest profile	Active/passive	Success rate	Number of Nodes
1	Sport	Active	90%	515
2	Dinning	Active	86%	518
3	Shopping	Passive	83%	522
4	POI			
5	Sport	Active	82%	558
	Shopping			
	POI			
	Dinning			
	Sport	Passive	77%	440
	Shopping			
	Dinning			

4.6 Summary

We set off to improve the interactivity and playability of virtual world and make the environment more enjoyable by employing agent technologies. In this chapter, we have focused on the issues and challenges for building learning agents in virtual world. To this end, we presented TD-FALCON, which enables an autonomous agent to adapt and function in a dynamic environment, as the model for implementing learning agents for

personal services in virtual worlds. Our experimental results and user case study have supported the validity of our agent systems.

Chapter 5

Learning Personal Agents with Adaptive Player Models

5.1 Introduction

User modeling is the user cognitive process in which the data on users occurred under a specific system environment are collected and analyzed to achieve the objective of the system more effectively. User modeling technique is being exploited diversely to analyze the profile information of users in the many fields such as games, educations, ubiquitous technologies and web mining.

Particularly in the game environment, since each player has a different preference of gameplay within a game, and the range of game playing capabilities between players can vary widely, “learning from player” has been recognized as a break through for game AI. The game captures the skills, weaknesses, preferences and other characteristics of that player, and the game AI makes decision of how to respond to the player based on these captured information.

In this chapter, we aim to enable more dynamic and personalized experience by employing player modeling technology in virtual worlds. We propose to incorporate adaptive user modeling with our reinforcement learning agents. Specifically, a two-channel fusion ART [1, 2], is exploited for learning player models during the players’ interaction. By formulating cognitive codes associating agent’s recommendation to user feedback, fusion ART learns direct player models of the users’ like and dislike.

We have developed techniques for adaptive player modeling in a 3-D virtual world called Youth Olympic Village (YOV) Co-Space. We have integrated adaptive user models

with learning personal agents which are designed to befriend human users and proactively offer them with personalized functions and services in the virtual environment [79]. Our experimental results show that we are able to learn player models that evolve and adapt with players during run time. Furthermore, the virtual tour guides with adaptive player models outperform those without player models. Although the idea of player modeling has been proposed before, most of these works do not work in conjunction with a reinforcement learning agent. Even few have attempted to provide any quantitative assessment. Our research work are motivated by such a consideration. This work has been published in [KangTan10b, KangTan10c].

5.2 Problem Formulation

To define the problem of adaptive player modeling, we start with a set of attributes $F = \{F_1, F_2, \dots, F_n\}$ and an additional set $V = \{V_1, V_2, \dots, V_m\}$, where F is the feature set which represents the patterns of players such as the player's behaviors, player's preference and traits, and the class set V represents the class of player's preferences. In this online supervised learning problem, the system should autonomously construct the player model and make prediction during run-time. At the model construction phase, the supervised problem involves constructing the training sets by a mapping $g : F_1 \times F_2 \times \dots \times F_n \rightarrow V$, such that there exists a minimal variability between $g(f_1, f_2, \dots, f_n)$ and the corresponding value v . Based on these training sets, when a new set of feature values, such as $f'_1, f'_2, \dots, f'_n$ is given, $g(f'_1, f'_2, \dots, f'_n)$ is used to represent the estimated value of the class variable v' . Alternatively, when a new class variable v'' is given, the desired set of feature values $f''_1, f''_2, \dots, f''_n$ should satisfy a minimal variability between $g(f''_1, f''_2, \dots, f''_n)$ and the corresponding value v'' . In this way, the system can either predict the players' intention given their patterns or the players' patterns given the intention.

5.3 Fusion ART

As shown in Figure 5.1, Fusion ART [77] is based on multi-channel Adaptive Resonance Associative Map (multichannel ARAM) [80], an extension of predictive Adaptive Resonance Theory (ART) networks. Whereas predictive neural network models, such as

ARTMAP network [78] and ARAM [81], learn multi-dimensional mappings between the input and output patterns, fusion ART formulates cognitive codes associating multi-modal patterns across multiple input channels. With the properties of self-adaptation, generalization and fast yet stable real-time learning, fusion ART makes a suitable building block for player modeling.

The generic network dynamics of fusion ART, based on fuzzy ART operations [78], is described as follows.

Figure 5.1: The architecture of fusion ART.

Input vectors: Let $\mathbf{S} = (s_1, s_2, \dots, s_n)$ denote the state vector, where $s_i \in [0, 1]$ indicates the value of sensory input i . Let $\mathbf{A} = (a_1, a_2, \dots, a_m)$ denote the action vector, where $a_p \in [0, 1]$ indicates the preference of a possible action p . Let $\mathbf{R} = (r, \bar{r})$ denote the reward vector, where $r \in [0, 1]$ is the reward signal value and $\bar{r}$ is given by $\bar{r} = 1 - r$. The whole input vectors are with complement coding.

Activity vectors: Let $\mathbf{x}^{ck}$ denote the F_1^{ck} activity vector for $k = 1, \dots, K$, where K is a positive integer. Let $\mathbf{y}$ denote the F_2 activity vector.

Weight vectors: Let $\mathbf{w}_j^{ck}$ denote the weight vector associated with the j th node in F_2 for learning the input patterns in F_1^{ck} for $k = 1, \dots, K$. Initially, F_2 contains only one *uncommitted* node. An *uncommitted* node is one which has not been used to encode any pattern and its weight vector contains all 1s.

Parameters: The fusion ART's dynamics is determined by choice parameters $\alpha^{ck} \geq 0$, learning rate parameters $\beta^{ck} \in [0, 1]$, contribution parameters $\gamma^{ck} \in [0, 1]$ and vigilance parameters $\rho^{ck} \in [0, 1]$ for $k = 1, \dots, K$.

The FALCON pattern processing cycle comprises of five key stages, namely code activation, code competition, activity readout, template matching, and template learning, as described below.

Code activation: A bottom-up propagation process first takes place in which the activities of the category nodes in the F_2 field are computed. Specifically, given the activity vectors $\mathbf{x}^{c1}, \mathbf{x}^{c2}, \dots, \mathbf{x}^{cK}$, for each F_2 node j , the choice function T_j is computed as follows:

$$T_j = \sum_{k=1}^K \gamma^{ck} \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_j^{ck}|}{\alpha^{ck} + |\mathbf{w}_j^{ck}|}, \quad (5.1)$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the norm $|\cdot|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code competition: A code competition process follows under which the F_2 node with the highest choice function value is identified. The system is said to make a choice when at most one F_2 node can become active after the code competition process. The winner is indexed at J where $T_J = \max\{T_j: \text{for all } F_2 \text{ node } j\}$.

When a category choice is made at node J , $y_J = 1$ and $y_j = 0$ for all $j \neq J$. This indicates a winner-take-all strategy.

Activity readout: The chosen F_2 node J performs a readout of its weight vectors into the input fields F_1^{ck} such that

$$x^{ck(new)} = x^{ck(old)} \wedge w_J^{ck}. \quad (5.2)$$

The resultant F_1^{ck} activity vectors are thus the fuzzy AND of their original values and their corresponding weight vectors.

Template matching: Before the node J can be used for learning, a template matching process checks that the weight templates of node J are sufficiently close to their respective input patterns. Specifically, resonance occurs if for each channel k , the *match function* m_J^{ck} of the chosen node J meets its vigilance criterion ρ^{ck} :

$$m_J^{ck} = \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck}|}{|\mathbf{x}^{ck}|} \geq \rho^{ck}. \quad (5.3)$$

If any of the vigilance constraints is violated, mismatch reset occurs in which the value of the choice function T_j is set to 0 for the duration of the input presentation. The search process then selects another F_2 node J until a resonance is achieved.

Template learning: Once a resonance occurs, for each channel ck , the weight vector $\mathbf{w}_J^{ck}$ is modified by the following learning rule:

$$\mathbf{w}_J^{ck(\text{new})} = (1 - \beta^{ck})\mathbf{w}_J^{ck(\text{old})} + \beta^{ck}(\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck(\text{old})}). \quad (5.4)$$

When an uncommitted node is selected for learning, it becomes *committed* and a new uncommitted node is added to the F_2 field. Fusion ART thus expands its network architecture dynamically in response to the input patterns.

Challenges for applying fusion ART include identifying the input for each channel and using the right parameters. The following sections will discuss these challenges in detail.

5.4 A Case Study on Learning Player Models in YOV Co-Space

5.4.1 Modeling Players

Considering the sets of information a personal agent can recommend, we define the player models consisting of six categories accordingly, namely, Preferred Hotel Star, Consumption Level, Favorite Type of Food, Favorite Type of Sports and Favorite Type of POI. The possible values with these attributes are shown in Table 5.1.

5.4.2 Learning Player Models

We adopt a two-channel fusion ART (Figure 5.2) for learning player models. Essentially, the model performs supervised learning through the pairing of the input patterns and teaching signals received from the virtual environment without involving human supervision and intervention. If an initial user profile is available, the model first creates cognitive nodes that associate the attributes specified in the player’s profile with positive reward signals. During play time, the player model learns the user’s like and dislike by formulating cognitive codes associating the attributes of the agent’s recommendations to users’ feedbacks. For example, if the agent recommends a Japanese restaurant “Dozo” and the user likes it, the attribute corresponding to Japanese in the player’s profile and the reward attribute are set to ‘1’ while other attributes are set to ‘0’. Fusion ART then

Table 5.1: Attributes and possible values in the state space of player model.

Attributes	Values
Preferred Hotel Class	5 Star, 4 Star, 3 Star and below
Favorite Type of Food	Chinese, Western, International, Taiwanese, Seafood, Indian, Mexican, Korean, Thai, Halal, Italian, Singapore, Japanese, French, Steamboat, Nonya and Peranakan, Vegetarian, Vietnamese
Consumption Level	Luxury, Medium, Budget
Favorite Type of Sports	Aquatics, Archery, Athletics, Badminton, Basketball, Boxing, Canoe-Kayak, Cycling, Fencing, Football, Gymnastics, Modern Pentathlon, Rowing, Sailing, Shooting, Table tennis, Triathlon, Volleyball, Weightlifting, Judo, Hockey, Equestrian, Handball, Taekwondo, Wrestling, Tennis
Favorite Type of POI	Church, Memorial, Mosque, Palace, Temple, Zoo
Favorite Type of Shopping	Children and Maternity, Electrical and Electronic, IT and Telecommunication, Art and Craft, Jewellery and Watch, Fashion and Accessory, Entertainment and Music, Leather good and Footwear, Health and Pharmacy and Toilet

encodes these patterns into a cognitive code. Performing supervised learning in real time, it is also able to adapt itself to the changes in the user preference patterns.

Figure 5.2: A two channel fusion ART architecture for supervised learning of user models.

During the learning mode, given the input space S and output space A , the system first performs code activation and code competition (as described in Section IV) to select a winner J . Before node J can be used for learning, a template matching process checks that the weight templates of node are sufficiently close to their respective input patterns. If the chosen node J violates its vigilance criterion, the search process then selects another F_2 node J under the revised vigilance criterion until a resonance is achieved. Once a node J is selected for firing, for each channel k , the weight vector is modified by the template learning rule.

In a predicting mode, fusion ART receives input patterns from one or more input fields and predicts the patterns in the remaining fields. The predicting process follows three key steps, namely, code activation, code competition, and activity readout. In case of more than one winners during the code competition process, these nodes perform a combined activity readout. Furthermore, the players' general interest could be inferred based on the assumption that the greater the number of player's preferences in one general category, the more the player is interested in this category. In this way, player's interest in the state space of personal agent (shown in Table 5.1) will be updated over run-time.

5.4.3 Input Space Representation

The input space of the player model consists of six sets of attributes, namely Preferred Hotel Class, Favorite Type of Food, Consumption Level, Favorite Type of Sports, Favorite Type of POI and Favorite Type of Shopping. A summary of these attributes together with the possible values is given in Table 5.1. All attributes adopt a binary encoding scheme in the state vector.

5.4.4 Output Space Representation

The player model is defined to simulate players with different background, personalities and interests. As a start, our action field only consists of two attributes: Like and dislike, represented using the binary encoding scheme.

5.5 Integrating Player Model and Personal Agent

The architecture of the overall recommendation agent, incorporating the personal agent, the search agent, and the player model, is shown in Figure 5.3. As illustrated in this framework, we incorporate the two-channel fusion ART based player model together with the personal agent and the search agent in recommending functions and services to the users. Specifically, the personal agent determines the appropriate type of services to recommend, such as hotel, restaurant, YOV, POI and shopping according to the current environment situation and user’s context. Fusion ART then infers the player model especially player special preference and updates the personal agent with the players’ current general interests during the interplay. By incorporating the personal agent with the adaptive user model, the system would be more adaptive to the players’ habits and eccentricity. Based on the output of the personal agent and the player model’s inferred result, the search agent works like a search engine that handles the retrieving of requested information from the database.

A typical example of how the various components work together is described as follows: When it is near dinner time, the personal agent suggests the service of providing the restaurant information to the player. Meanwhile, the player model indicates that this player prefers Japanese cuisine. Hence, the query of Restaurant and Japanese will be passed to the search agent, and Japanese restaurants with the highest matching scores will be recommended to the player. As the player requests for different restaurants, other cuisine types will be learnt by player model and updated as the player’s interest after competing with other categories of general interest. Hence, during the following play time, the personal agent will give updated recommendations accordingly.

5.6 Empirical Experiments

Our previous study [79] showed that the TD-FALCON based personal agents can adapt to the environment and improve their performance in real time. In this dissertation, we investigate whether we are able to learn player models that evolve and adapt with players during run time using fusion ART. Furthermore, we wish to evaluate the performance of the personal agents with adaptive player models in comparison with those without player

Figure 5.3: The architecture of recommendation agent.

models. Based on these considerations, we conduct three sets of experiments, namely experiments with static player profile, experiments with evolving player profile and experiments without player registered profile. Player registered profiles are the personal information captured by the system during user registration. Learning user models with and without initial profiles provide different challenges to fusion ART.

For learning user models, we build a fusion ART network comprising 130 nodes in the input space field and two nodes in the output field. Moreover, in order to evaluate our system performance empirically, we design an automatic test procedure using synthetic player models, simulating the player's feedback to the agent's service based on a set of player logic. By matching the agent's recommendations with the player's expectation, we are able to compute reward signals that can be used to guide the agent in learning. A sample set of player logic is given in Table 5.2 as an illustration, where x is a value of Player's Interest in Table 5.1 and P is a set of the player's preference corresponding to the values in Table 5.1. Given the current situation, if the output recommendation matches the expected recommendation, a reward of value 1 will be given. Otherwise, a reward of value 0 will be given.

As the main role of personal agents is to provide services to the players, we evaluate their performance by the accuracy of the service provided to the players over a period of

Table 5.2: Logic for generating rewards based on a synthetic player model P.

IF	Time = Lunch or Dinner Time, Activity != Dinning, Request = Unknown,
EXPECT	Recommend restaurant with attributes $a_i \in P$
IF	Time = Lunch or Dinner Time, Activity = Dinning, Request = Unknown, Interest contains x
EXPECT	Recommend x with attributes $a_i \in P$
IF	Time = Bed Time, Request = Unknown
EXPECT	Recommend Hotel with attributes $a_i \in P$

time.

5.6.1 Experimenting with Static Player Profile

In this set of experiment, the player profile will not evolve during run-time. Fusion ART uses a default set of parameter values as listed in Table 5.3. The recommendation accuracy with static user profile averaged at 100-trial intervals over 3000 trials are shown in Figure 5.4. The results, obtained after averaging across five sets of experiments, generally indicate that the agents are able to gradually improve their performance through user feedback continuously over time. We can observe that when there is no change in the player profile, the performance is largely similar at all interaction cycles. This is because player registered profiles are beneficial to player model. They can be considered as the prior knowledge to initialize the player model.

Table 5.3: Fusion ART Parameter setting for Player Models.

Fusion ART Parameters	Values
Choice parameter(α^{c1}, α^{c2})	0.1, 0.1
Learning rates(β^{c1}, β^{c2})	1.0, 1.0
Contribution parameter(γ^{c1}, γ^{c2})	0.5, 0.5
Baseline vigilance parameter(ρ^{c1}, ρ^{c2})	1.0, 0.0

Figure 5.4: Recommendation accuracy of recommendation agents with registered player profile.

5.6.2 Experimenting with Evolving Player Profile

In this set of the experiments, we evolve the player profile during run-time to investigate how the agents may cope with a shift in player’s interests. After 1500 learning trials, we changed 20% of the interest in synthetic player model by shifting the player’s preference from Japanese to seafood and steamboat and from Aquatics to Sailing and Table Tennis. Moreover, we add in new preferences after 1500 learning trials to shift the user’s general interest from dinning to shopping.

As shown in Figure 5.5, the recommendation accuracy of both systems are largely similar over the first 1500 learning trials, which are consistent with the previous set of experiments. As the target player profile evolves after 1500 learning trials, both of them suffer a critical drop in performance at the next 100 trials. However, they all manage to improve the performance afterwards. Even without the learning player model, due to the learning ability of the personal agent, it can adapt and give general recommendations according to the feedbacks from the players. However, as the player profile has changed, the recommendation agent without a player model does not have the capability of adapting to the new interest and preferences. Hence, the recommendation

agent with learning player model significantly outperform its counterpart after the player profile evolves.

Figure 5.5: Recommendation accuracy of recommendation agent with evolving player profile.

5.6.3 Experimenting without Player Registered Profile

All the above mentioned experiments rely on the player’s registered profile. However, in a realistic environment, players may not always fill in their information. They may log into the Co-Space by using a “guest” account. We note that this is more challenging because it requires the capability of learning from zero knowledge. In this set of experiments, we evaluate such a scenario. During the experiment, the player’s preference exactly follows the setting of last experiment. The performance of learning player model, in terms of prediction accuracy, can be found in Figure 5.6.

Since there is no more “pre-inserted hints”, recommendation agent without a learning player model suffers a more significant drop of more than 50% comparing with that in the previous set of experiments. In contrast, fusion ART, despite without the player registered profile, quickly learns from zero knowledge and adapts to the player. Even after significant changes of player’s interest, it manages to regain an accuracy of more than 80% eventually.

Figure 5.6: Recommendation accuracy of recommendation agent without registered player profile.

5.7 Summary

The main objective of our work is to make the players' experience in virtual worlds more intense and personalized by employing learning player model technologies. In this chapter, we have focused on the capacity of agents to adapt over time to the habits and eccentric of a particular player in virtual world. To this end, we present a personal agent based on FALCON, which enables an autonomous agent to adapt and function in a dynamic environment, and augmented it with an adaptive player model for more personalized services in virtual worlds. Our experimental results have so far supported the validity of our multi-agent approach.

Chapter 6

Self-Organizing Cognitive Models for Virtual Human

6.1 Introduction

In the previous chapters, we have addressed two key challenges of in building intelligent virtual human in virtual world, namely adaptive learning ability and user modeling. In this chapter, we present a fully connectionist architecture that realizes the key features of realistic virtual humans, namely autonomy, interactivity and personality in an integrated way. Extending from a family of self-organizing neural networks known as Adaptive Resonance Theory (ART) [15], the brain-inspired agent architecture named Agent model Integrating Autonomy, Interactivity and Personification (AIAIP) integrates goal-directed autonomy, natural language interaction, proactiveness, situation awareness and human-like personality. In this way, these agent-based virtual humans are living a more realistic life, going to classes, having dinner with friends and chit-chatting with the others. This work has been published in [Kang12a, Kang12c].

6.2 A Neural Network Instantiation of Virtual Humans

6.2.1 Architecture

In this section, we present a realization of the proposed cognitive agent model based on self-organizing neural networks. As shown in Figure 6.1, the AIAIP model comprises

multiple interconnecting fusion ART network modules, each represents a component in the cognitive agent model.

Figure 6.1: A neural network implementation of the integrated cognitive agent model.

Perception: Let $\mathbf{I}^{c1} = \mathbf{S}$ denote the state vector, where $I_i^{c1} \in [0, 1]$ indicates the value of an input signal received by the agent. With complement coding, the input vector $\mathbf{I}^{c1}$ is augmented with a complement vector $\bar{\mathbf{I}}^{c1}$ such that $\bar{I}_i^{c1} = 1 - I_i^{c1}$.

As the perception buffer receives input signals obtained through sensors from the external environment, some preprocessing of the input signals may be necessary in actual embodiment of the agent. Also, in the more general case, the content of the state vector may be further modified and augmented by the belief of the agent.

An example of the agent belief may be implemented as a two-channel fusion ART network that consists of an input field F_1^{bi} for representing the current state, an output field F_1^{bo} for representing the modified state after reasoning, and a cognitive field F_2^b for encoding the inferencing rules for reasoning. Given the current state vector $\mathbf{S}$, an inference rule is selected based on its similarity with the rule's condition. If the inference rule satisfies the vigilance criterion, the consequents of the rule is added to form the new state $\mathbf{S}'$. The reasoning process may be repeated until no applicable inference rule is identified.

Desire: The desire module maintains and manages the agent's goals. It may be implemented as a one-channel fusion ART network that consists of an input field F_1^d for representing the goals to be achieved and a cognitive field F_2^d for encoding the active goals. Specifically, each committed node in F_2^d field is used to encode the target state of an active goal. During system initialization, the goal vectors are presented to the desire module for encoding. Hence, each goal j is encoded by a weight vector w_j^d associated to the F_2^d node j .

As an input to the inference engine, let $\mathbf{I}^{c2} = \mathbf{D}$ denote the desire vector, where $I_k^{c2} \in [0, 1]$ indicates the strength of a goal k pursued by the agent.

Personality: Based on the Five Factor Model (FFM) (also known as the OCEAN model) [82], the personality of the agent can be defined as a tuple $\mathbf{P}_a = (\mathcal{O}_a, \mathcal{C}_a, \mathcal{E}_a, \mathcal{A}_a, \mathcal{N}_a)$. Specifically, $\mathcal{O}_a$ or *openness* is the preference of agent a towards novelty or variety; $\mathcal{C}_a$ or *conscientiousness* is the level of agent a towards disciplined, organized, or responsible in goal achievement; $\mathcal{E}_a$ or *extraversion* is the agent a degree of assertiveness and talkativeness; $\mathcal{A}_a$ or *agreeableness* is the level of agent a in being kind, cooperative, or helpful towards others; and $\mathcal{N}_a$ or *Neuroticism* is the agent level of anxiety, emotional stability, and/or calmness. By giving a weighage to each dimension, a unique personality can be formed by a combination of the traits.

As an input to the inference engine, let $\mathbf{I}^{c3} = \mathbf{P}$ denote the personality vector, where $I_k^{c3} \in [0, 1]$ indicates the strength of a personal attribute k by the agent.

Action: Let $\mathbf{I}^{c4} = \mathbf{A}$ denote the action vector, where $I_k^{c4} \in [0, 1]$ indicates the value of an output command signal produced by the agent. The action vector $\mathbf{I}^{c2}$ is also typically augmented with a complement vector $\bar{\mathbf{I}}^{c2}$ such that $\bar{I}_k^{c2} = 1 - I_k^{c2}$. For decision making, the action K with the maximum I_k^{c2} value, is selected for performance.

Reward: Let $\mathbf{I}^{c5} = \mathbf{R}$ denote the reward vector, where $I_1^{c5} \in [0, 1]$ indicates the value of the internal feedback signal computed by the agent. The reward $\mathbf{I}^{c3}$ is also typically complement coded for ease of computation.

With a specific goal in mind, upon deriving the current state, the goal attainment function GA computes the match value between the current state vector $\mathbf{S}$ and the target goal vector $\mathbf{D}$. If the current state matches perfectly with the goal target, $A^d = 1$, or else GA can be a normalized difference between the two vectors. By matching the current

state with the active goal, the desire module computes how well the agent has done towards its desired target. The computed degree of goal attainment in turn serve as reward signals to the feedback field of the inference engine.

6.2.2 Network Dynamics

Considering the dynamic environment and the diverse set of factors involved in decision making, a self-organizing neural model is employed, that integrates multi-modal input patterns and discovers action policy through evaluative reinforcement feedback.

The inference engine of AIAIP employs a five-channel Fusion ART architecture comprising a category field F_2 and five input fields, namely, a sensory field F_1^{c1} for representing current states, F_1^{c2} representing the desire, F_1^{c3} for representing personality, F_1^{c4} for representing a motor field and a feedback field F_1^{c5} for representing reward values.

Activity vectors: Let $\mathbf{x}^{ck}$ denote the F_1^{ck} activity vector. Let $\mathbf{y}$ denote the F_2 activity vector. Upon input pattern presentation, $\mathbf{x}^{ck} = \mathbf{I}^{ck}$ for $k = 1, \dots, 5$

Weight vectors: Let $\mathbf{w}_j^{ck}$ denote the weight vector associated with the j th node in F_2 for learning the input patterns in F_1^{ck} . Initially, F_2 contains only one *uncommitted* node. An *uncommitted* node is one which has not been used to encode any pattern and its weight vector contains all 1s. Different weight vectors could be predefined to enable initial decision making to be reasonable and fast.

Parameters: The fusion ART's dynamics is determined by choice parameters $\alpha^{ck} \geq 0$, learning rate parameters $\beta^{ck} \in [0, 1]$, contribution parameters $\gamma^{ck} \in [0, 1]$ and vigilance parameters $\rho^{ck} \in [0, 1]$ for each of the input field k for $k = 1, \dots, K$.

Code activation: Given the activity vectors $\mathbf{x}^{ck}$, for each F_2 node j , the choice function T_j is computed as follows:

$$T_j = \sum_{K=1}^5 \gamma^{cK} \frac{|\mathbf{x}^{cK} \wedge \mathbf{w}_j^{cK}|}{\alpha^{cK} + |\mathbf{w}_j^{cK}|}, \quad (6.1)$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the norm $|\cdot|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

Code competition: The system is said to make a choice when at most one F_2 node can become active after the code competition process. The winner is indexed at J where

$$T_J = \max\{T_j : \text{for all } F_2 \text{ node } j\}. \quad (6.2)$$

When a category choice is made at node J , $y_J = 1$ and $y_j = 0$ for all $j \neq J$. This indicates a winner-take-all strategy.

Activity readout: The chosen F_2 node J performs a readout of its weight vectors into the input fields F_1^{ck} such that

$$\mathbf{x}^{ck(new)} = \mathbf{x}^{ck(old)} \wedge \mathbf{w}_J^{ck}, \quad (6.3)$$

The resultant F_1^{ck} activity vectors are thus the fuzzy AND of their original values and their corresponding weight vectors.

Template matching: Resonance occurs if for each channel k , the *match function* m_J^{ck} of the chosen node J meets its vigilance criterion ρ^{ck} :

$$m_J^{ck} = \frac{|\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck}|}{|\mathbf{x}^{ck}|} \geq \rho^{ck}. \quad (6.4)$$

If any of the vigilance constraints is violated, mismatch reset occurs in which the value of the choice function T_J is set to 0 for the duration of the input presentation. The search process then selects another F_2 node J until a resonance is achieved.

Template learning: Once a resonance occurs, for each channel ck , the weight vector $\mathbf{w}_J^{ck}$ is modified by the following learning rule:

$$\mathbf{w}_J^{ck(new)} = (1 - \beta^{ck})\mathbf{w}_J^{ck(old)} + \beta^{ck}(\mathbf{x}^{ck} \wedge \mathbf{w}_J^{ck(old)}). \quad (6.5)$$

A generic flow of the fusion-ART based agent model is presented in Algorithm 2. It incorporates Temporal Difference (TD) methods to estimate and learn value functions of the state-action pairs $Q(s, a, d, p)$ that indicates the goodness for a learning system to take a certain action a in a given state s , with desire d and personality p . Such value functions are then used in the action selection mechanism, also known as the *policy*, to select an action with the maximal payoff. The algorithm selects an action with the maximal Q-value in a given state s , with desire d and personality p , by enumerating and evaluating each available action a by presenting the corresponding state vector $\mathbf{S}$, desire vector $\mathbf{D}$, and personality vector $\mathbf{P}$, to fusion ART.

Algorithm 2: Dynamics of the fusion ART based Inference Engine

```
1 Initialize the Fusion ART network;  
2 Given the current state  $s$ , desire  $d$ , personality  $p$  ;  
3 while  $s$  is not a terminal state do  
4   for each available action  $a$  in the action set  $\mathbf{A}$ , predict the value of the action  
    $Q(s,a)$  by presenting the corresponding field and action vectors to Fusion-ART;  
5   Based on the value functions computed, select an action  $a$  from  $\mathbf{A}$  following an  
   action selection policy;  
6   Perform the action  $a$ , observe the next state  $s'$ , personality  $p'$  and receive a  
   reward  $r$  (if any) from the environment;  
7   Estimate the value function  $Q(s, a, d, p)$  following a temporal difference  
   formula given by  $\Delta Q(s, a, d, p) = \alpha T D_{err}$ ;  
8   Present the corresponding state, desire, personality, action, and reward  
   (Q-value) vector, namely  $\mathbf{S}$ ,  $\mathbf{D}$ ,  $\mathbf{P}$ ,  $\mathbf{A}$ , and  $\mathbf{R}$ , to Fusion ART for learning;  
9   Update the current state by  $s = s'$ ;
```

6.3 Natural Language Understanding and Natural Language Generation

The previous sections have presented the architecture of virtual human and the internal process of reasoning and making decisions. These decisions include non-verbal behaviors as well as verbal behaviors. While the commands for doing non-verbal behaviors are relatively easy to perform, the commands for verbal behaviors are not that straightforward to perform. Additional module is required to deal with the natural languages, especially those informational dialogues.

6.3.1 Existing Work

Before discussing our techniques for dealing with dialogue, we review the most commonly used methods for chat bot, namely the pattern matching approach. The core of the architecture is the stimulus-response algorithm, which determines the reaction to a user input via a database containing a fixed set of pattern-template pairs. The pattern can be seen as a surface string optionally enriched with regular expression syntax which can be matched to the user input. If an input successfully matches with an existing pattern, the hard linked template is delivered as the answer to the user. As shown in Figure 6.2, it

contains a component for input analysis which does some string cleaning on the surface structures coming from the user, the dialogue core which uses pattern matching to select the next utterance, and an output component which selects the next output. An example of a typical process of pattern matching mechanism is shown in Table 6.1.

Figure 6.2: A typical pattern matching approach for dialogue generation.

Table 6.1: Example of a typical pattern matching mechanism

Steps	Actions done	result
Input	The raw input	Hey man, how are you doing today?
Input cleaning	Removal and substitution of characters and words	How are you doing today
Pattern matching	Search a pattern which matches the cleaned input in the database	⟨ HOW ARE YOU DOING TODAY ⟩
Pattern matching	Get the matching answer templates	⟨ I'M FINE ⟩⟨ I'M DOING GOOD! ⟩
Output selection	Select one of the answers	I'm doing good!

Although it is possible to adopt the pattern matching mechanism for processing informational questions, there are several limitations. First, the pattern matching mechanism requires pre-defining a sufficient number of patterns for effective question answering. For conversational utterances, we can re-use the resources of the existing projects. However, for different projects, all the informational questions are specialized and need to be defined. As the number of informational questions to be asked is huge, the use of the pattern matching mechanism becomes a tedious work. Second, the pattern matching mechanism simply finds a matched pattern for question answering. It does not support more advanced functions such as capturing the players intentions and interests. Consequently, the agents can only perform simple actions.

The traditional methods for processing informational questions is based on the notion that the meaning of the utterances can be captured in formal structures such as first order logic [83] or semantic network [84], which is called meaning representations. After

getting the meaning representations of the questions, they substitute possible objects of the knowledge base into the meaning representation, and select objects that make the representations be true as the answers of the questions. While the traditional methods are shown to be an effective method for this problem, the development of such methods will require huge human efforts. For example, the development of the IBM Watson system [85] has involved a lot of researchers. It is hard to adopt the traditional methods.

6.3.2 Extending Cognitive Agent Models

In view of the limitation of the pattern matching mechanism, we propose to incorporate the *Chat Understanding Module* and the *Natural Language Generation Module* with the previous presented model. As shown in Figure 6.3, the integrated agent model consists of a *Perception Module* receiving situational signals from the environment through a set of sensory Application Programming Interfaces (APIs) and an *Action Module* for performing actions through the various actuator APIs. If the sensory signals involve a text input, the *Chat Understanding Module* interprets the text for the player's intention. The outputs of *Situational Assessment* and *Chat Understanding Modules* thus serve as the perception buffer. The *Inference Engine* then identifies the most appropriate action, in accordance to the desire and personality of the virtual agent. The decision of the *Inference Engine* forms the *Motion Buffer*. For actions involving a verbal response, the *Natural Language Generation Module* translates the chosen response into natural text for presentation, by tapping the *Belief*, which includes the declarative knowledge of self, players, and environment, as well as the procedural knowledge of goal-oriented rules, which guide an agent in fulfilling the goals, and the social rules for generating socially appropriate behavior.

Chat Understanding: Chat understanding delivers a representation of the users input which can be used by the inference engine. It includes three steps:

- (i) Input cleaning: Delete and substitute unwanted characters, with potentially correction of common typing and grammar errors.
- (ii) Segmentation: Segmentation of the user input in single utterances.

Figure 6.3: The detailed implementation of the integrated cognitive agent model.

- (iii) Word-level analysis: Processing of the words part of speech and morphological information, such as Named Entity Recognition(NER) [86].

Natural Language Generation: As the Inference engine transfers the given information from *Chat Understanding* into a description of the meaning of the utterance, the *Natural Language Generation Module* constructs natural language messages encapsulating this information. Firstly, we classify the meaning of the utterance into corresponding semantic categories to get the intention of the questions. Then, we generate answers by looking into the *Belief* that contain keywords of the question. We call it IR based QA approach. An example to illustrate this approach is given in Figure 6.4, where we want to answer *where is Canteen 2?* Firstly, we find the intent of this question is about the location of some places. Then, we use the keywords of the question to search the *Belief*, and retrieve the answer. To increase the flexibility, we are using *Template Filling* method so that pre-encoded text snippets may contain variable slots in which different content can be inserted dynamically. These templates can produce one and the same output with slight variations and therefore provide more flexibility to the system.

Figure 6.4: The process of natural language understanding and generation.

6.4 System Properties

AIAIP performs a myriad of cognitive functions, including recognition, prediction and learning, in response to a continual stream of input signals received from multiple pattern channels. As a result, an agent makes decisions not only based on the situational factors perceived from the environment but also her mental states characterized by desire and personality.

6.4.1 Human-Like Properties

Through the interplay of the various modules, the agent model as presented is able to exhibit certain levels of human-like characteristics. The key properties are reviewed as follows.

Property 6.1 (Interactivity) *AIAIP is capable of exhibiting interactivity with its environment.*

Given a change in the situation of the environment or an user input, the input is encoded in the state vector in the perception module. Consequently, the inference engine selects an appropriate action using the choice function. As such, AIAIP is designed to provide a response to the input stimulus, realizing interactivity.

Property 6.2 (Autonomy) *AIAIP is capable of producing an action without explicit input from the external world.*

By maintaining internal goals as desire, AIAIP constantly evaluates the current state against the target state and performs actions autonomously, without explicit instructions.

Property 6.3 (Proactiveness) *AIAIP is able to make decision and performs an action based on its environment and internal goals, without explicit input from its human player or other agents.*

With the internal goals and belief of the world, AIAIP is able to make decision and performs an action without explicit input from its human player or other agent.

Property 6.4 (Situation Awareness) *AIAIP is aware of its environment and behaves accordingly.*

Given the situation of the environment represented as a state vector in the perception module, AIAIP is aware of its environment and makes decision accordance to the current situation.

Lemma 6.1 (Rationality) *Given the current desire, personality and perception at a specific time, AIAIP always selects an action with the maximal expected reward.*

Proof: (By Contradiction) Suppose AIAIP selects a F_2^c node J and there exists another F_2^c node K of which the weight vector $\mathbf{w}_K^{c1}$ is equal or more similar to $\mathbf{S}$ than $\mathbf{w}_J^{c1}$, the weight vector $\mathbf{w}_K^{c2}$ is equal or more similar to $\mathbf{D}$ than $\mathbf{w}_J^{c2}$, the weight vector $\mathbf{w}_K^{c3}$ is equal or more similar to $\mathbf{P}$ than $\mathbf{w}_J^{c3}$ and the weight vector $\mathbf{w}_K^{c5}$ encodes a equal or higher reward value than $\mathbf{w}_J^{c5}$. As $\mathbf{w}_K^{c1}$ is equal or more similar to $\mathbf{S}$ than $\mathbf{w}_J^{c1}$, we derive that

$$\frac{|\mathbf{x}^{c1} \wedge \mathbf{w}_K^{c1}|}{\alpha^{c1} + |\mathbf{w}_K^{c1}|} \geq \frac{|\mathbf{x}^{c1} \wedge \mathbf{w}_J^{c1}|}{\alpha^{c1} + |\mathbf{w}_J^{c1}|} \quad (6.6)$$

As $\mathbf{w}_K^{c2}$ is equal or more similar to $\mathbf{D}$ than $\mathbf{w}_J^{c2}$, we derive that

$$\frac{|\mathbf{x}^{c2} \wedge \mathbf{w}_K^{c2}|}{\alpha^{c2} + |\mathbf{w}_K^{c2}|} \geq \frac{|\mathbf{x}^{c2} \wedge \mathbf{w}_J^{c2}|}{\alpha^{c2} + |\mathbf{w}_J^{c2}|} \quad (6.7)$$

As $\mathbf{w}_K^{c3}$ is equal or more similar to $\mathbf{P}$ than $\mathbf{w}_J^{c3}$, we derive that

$$\frac{|\mathbf{x}^{c3} \wedge \mathbf{w}_K^{c3}|}{\alpha^{c3} + |\mathbf{w}_K^{c3}|} \geq \frac{|\mathbf{x}^{c3} \wedge \mathbf{w}_J^{c3}|}{\alpha^{c3} + |\mathbf{w}_J^{c3}|} \quad (6.8)$$

Likewise, as $\mathbf{w}_K^{c5}$ encodes a equal or higher reward than $\mathbf{w}_K^{c5}$, we have

$$\frac{|\mathbf{x}^{c5} \wedge \mathbf{w}_K^{c5}|}{\alpha^{c5} + |\mathbf{w}_K^{c5}|} \geq \frac{|\mathbf{x}^{c5} \wedge \mathbf{w}_J^{c5}|}{\alpha^{c5} + |\mathbf{w}_J^{c5}|} \quad (6.9)$$

By equation 6.1, the above conditions imply that $T_K^c \geq T_J^c$, which means node K should be selected by AIAIP instead of node J (Contradiction).

Property 6.5 (Personification) *AIAIP exhibits elements of personification by incorporating personal attributes in the decision making process.*

By incorporating personal attributes into one input channel of the inference engine, AIAIP is able to take into consideration personality during the decision making process. Specifically, the activity readout to the action field is consistent with the personality of the agent. Consequently, different personalities may lead to very different actions and behaviour even with the same situation.

6.4.2 Further Discussion

In addition to the human-like properties discussed in the previous sections, the design of AIAIP is consistent with the basic principles, as outlined by Pfeifer and Scheier [87], towards designing intelligent systems. These include the principle of sensory-motor coordination, ecological balance, the value principle, parallel and loosely coupled processes and cheap design.

The principle of sensory-motor coordination: In the AIAIP model, all intelligent behavior, such as perception, categorization, memory, are conceived as a sensory-motor coordination that serves to structure the sensory input and motor output.

Ecological balance: The “complexity” of the agent has to match the complexity of the task environment. In particular, given a certain task environment, there has to be a match among the complexity of sensors, motor system, and neural substrate. Since the agent makes decisions on the situational input, its internal mentor state (desire, intention,

personality), and it learns according to human's feedback or other agents' feedback, it is able to match and adapt to the surrounding circumstances.

The value principle: Upon the agent's action, the environment may provide evaluative feedback signals on the agent's goal attainment values. By comparing goal's attainment values and the targeted goals, a Q-value can be computed which can be learned by the agent as the value for performing the past actions in a specific situation. Therefore, the virtual human is equipped with a value system and with mechanisms for self-supervised learning employing principles of self-organization.

Parallel, loosely coupled processes: AIAIP interacts with the environment based on a large number of parallel, loosely coupled processes that run asynchronously and are connected to the agent's sensory-motor apparatus. The central processing module *inference engine* takes into considerations various inputs, such as desire, intention, personality, perception, context during sensing, and then makes decisions on an action to perform.

Cheap design: Since AIAIP is designed to learn from experience (such as the feedbacks), it doesn't require pre-programming of the agent's behaviour and actions under every circumstances. Hence, it is considered as a cheap design in the sense that the model reduces the developers' work on scripting for every possible situation for the agent.

6.5 Summary

For creating realistic virtual humans in the virtual world, we have proposed a cognitive agent architecture that integrates goal-directed autonomy, natural language interaction and human-like personality. Extending from a family of self-organizing neural models, the agent architecture maintains explicit mental representation of desires, personalities, self-awareness, situation awareness and user awareness.

In the next chapter, we will present a case study based on the model we presented in this chapter.

Chapter 7

Case Study and Evaluation of Agent based Virtual Human

7.1 Introduction

As a showcase of our work, we have developed a Co-Space that models the Nanyang Technological University (NTU) campus. We have created realistic 3D models of almost all locations in NTU and populated the Co-Space with avatars of human appearance and body proportion. We have deployed agent based virtual humans into NTU Co-Space.

In order to evaluate the effects of virtual humans on user experience, in this chapter, we have done a formal user case study where three NTU Co-Spaces with three different types of virtual agents in the form of avatars are developed. The first environment (E1) are populated with virtual humans that are able to show and display static messages to users but do not have the capability to interact with the users. In the second environment (E2), virtual humans are designed as embodied conversational agent using AIML [16] technology. The third environment (E3) is populated with intelligent autonomous virtual human with our proposed agent architecture. 58 students have been randomly assigned to one of the three environments. The scenario for them is that they are looking for an overseas university for an exchange program and are visiting NTU Co-Space to help them in making the decision. Each subject will be asked to complete a quest in the form of a mini-game, where they will experience the key places of the NTU campus through the quest. Before they started, they were asked to fill a pre-questionnaire where students' demographics information, computer experience and 3D virtual world skill are captured.

After complete the quest, students were required to fill out a post-questionnaire to assess their experience. It captures ten various aspects of user experience including telepresence, social presence, perceived interactivity, 3D virtual world skill, perceived usefulness, flow, enjoyment, behavior intention. The result of this study has mostly supported the validity of our virtual human systems. To the best of our knowledge, this could be one of the few systematic empirical studies that has assessed the effect of intelligent virtual human in the virtual world on user experience. This work has been published in [Kang12b, KangTan13].

7.2 A Case Study on NTU Co-Space

Co-Space refers to interactive virtual environment modelled after the real world in terms of look-and-feel, functionalities and services. We have developed a 3D virtual world world named Nanyang Technological University (NTU) Co-Space populated with virtual human characters. It is implemented using the Unity3D [88], a 3D game engine that can be deployed into different platforms, including Windows, MacOS, popular game consoles, and mobile devices. The Co-Space can also be embedded in a web page to be easily accessed using typical web browsers such as Internet Explorer, Chrome, Safari and Firefox.

The overall architecture of the NTU Co-Space is depicted in Figure 7.1. At the back-end, two application servers, Apache Tomcat and SmartFox, support the multi-user environment. A dedicated database is used for the storage and retrieval of environmental and content data.

A video clip of the NTU Co-Space can be viewed on You Tube (<http://www.youtube.com/watch?v=bYIthOYjrxw>). During the live demo, multiple players will be able to log in and experience the NTU Co-Space. As illustrated in Figure 7.2, a player may choose to roam around the campus on his/her own for self-discovery or play a mini-game, called *Amazing Quest*, which will bring the player to experience the key places of the NTU campus through a fun and interactive journey. Specifically, the player will visit the five key landmarks on campus, with the accompaniment of a virtual character. Besides showcasing the intelligent agent technologies, the NTU Co-Space can be played by any prospective students and general public who are keen to know NTU.

Figure 7.1: System architecture of NTU Co-Space.

7.2.1 Embodiment of Agent Architecture

We assume the virtual agent has a total of four goals: (1) To stay healthy, (2) To graduate, (3) To make friends and (4) To be wealthy. He adopts the model FFM for his personalities.

Sample knowledge of environment and npc are illustrated in Table 7.1 and Figure 7.3. Samples of interaction and social rules are given in Table 7.2 and Table 7.3 respectively.

Table 7.1: Knowledge of campus.

Knowledge of Campus
(Campus, Name, NTU)
(Campus, Location, western part of Country)
(Campus, Number of colleges, 4)

The input state of the agent consists of three categories of context factors: Environmental context, virtual human context and player context. The environmental context

Figure 7.2: A player touring the NTU Co-Space.

About me

- (Self, Name, Isabella)
- (Self, Type, NPC)
- (Self, Sex, Female)
- (Self, Age, 22)
- (Self, Identity, College Student)
- (Self, Course, Computer Engineering)
- (Self, School, SCE)
- (Self, Year of Study, Fourth Year)
- (Self, Address, Hall 4)

Isabella

Figure 7.3: An example of agent profile model.

Table 7.2: A sample interaction rule.

IF	Personality is Agreeable
AND	Player ask for guidance to X
AND	No immediate planned activity
THEN	Guide player to X

Table 7.3: A sample social rule.

IF	Personality is Agreeable
AND	Player initiates conversation
AND	Distance is far
THEN	Walk towards the player

consists of global factors, such as time and location. The virtual human context comprises six types of attributes, including desire, personality, intention, previous performed action, natural language understanding and self model. The player context includes attributes such as player model, whether the player is approaching, whether the player starts a conversation, and whether the player requests something. A summary of these attributes together with the possible values is given in Table 7.4. All attributes adopt a binary encoding scheme in the state vector.

The virtual humans are designed to inhabit in the virtual world, behave autonomously, and interact with players and other virtual humans. The action field thus consists of three categories, namely chat related behaviors, social behavior and other behaviors. A summary of these attributes together with the possible values is given in Table 7.5.

7.3 Theoretical Background on User Experience

In recent years, the use of interactive online applications has become an integral part of everyday life, the exclusive usability test is no longer sufficient. Consequently, User Experience (UX) and its evaluation has been shifted to a more essential role in product or service development. In this section, we will discuss the definition of UX and the underlying constructs of evaluating UX.

7.3.1 Definition of UX

There are various definitions of UX, but not a universal one [89].

Table 7.4: Attributes and possible values in the state space.

Attributes	Values
Time	Morning, lunch Time, Afternoon, Dinner Time, Evening, Bed Time
Location	LT, Canteen, Shop, Hall, Library, Sports Facility, Schools, Admin Building, Banks, Other Building
virtual human's previous action	Approach, Guide a tour, Follow Schedule, Wandering, Start a conversation, End the conversation, Greeting, Talk about player's interest, Talk about campus, Offer help to player, Describe nearby landmark, Ask what player means, Ask what player want to know/do, Rephrase the question, Reply player's request, Provide suggestion, Continue previous conversation,
NLU	Type, Subject, Object, Modifier, Verb
Player	New/old player, Personality, Interest,
Model	Interaction preference, Goal
Agent	Personality, Interest,
Model	Interaction preference, Goal
Player activity	Wandering, Approach virtual human, Start Conversation, Make request, following guidance, respond to virtual human, give positive/negative/neural feedback to virtual human

Table 7.5: Categories and attributes in the action space.

Categories	Attributes
Non-Verbal Behavior	Approach, Guide a tour, Follow Schedule, Wandering
Verbal Behavior	Start a conversation, End the conversation, Greeting, Offer help to player, Talk about campus, Talk about player's interest, Describe nearby landmark, Ask what player means, Ask what player want to know/do, Continue previous conversation, Rephrase the question, Reply player's request, Provide suggestion

In ISO 9241-210, UX is defined as “a person’s perceptions and responses that result from the use or anticipated use of a product, system or service” [90]. According to this definition, user experience includes all the users’ emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors and accomplishments that occur

before, during and after use. It also list three factors that influence user experience: system, user and the context of use.

Jakob Nielsen and Don Norman summarize that UX encompasses all aspects of the end-user's interaction with the company, its services, and its products [91]. It emphasises that the products or services should be a joy to own, a joy to use. True user experience goes far beyond giving customers what they say they want, or providing checklist features.

According to Jim Miller, user experience “encompasses much more than the traditional 'user interface' issues, such as screen design and command structure. Rather, it's a broad collection of user-centric issues that cut through the full extent of a project” [92].

These well-known definitions are various, but they all agree that it is more than just a system's or service's usefulness and usability. Specifically, they focus more on the nature of UX: UX is affected by the users psychological issues like affectivity, emotions, cognition and performance [89]. Moreover, it is also about their interaction with products and services.

7.3.2 Related Constructs

We propose to use the following constructs to evaluate UX.

Telepresence: Telepresence refers to a set of technologies which allow a person to feel as if they were present, to give the appearance of being present, or to have an effect, at a place other than their true location. By employing intelligent agents into the virtual world, we think the feeling of telepresence will be enhanced.

Social Presence: Social-presence, as a subcomponent of presence, is composed of social richness, social realism and co-presence [93]. We believe that by employing intelligent virtual agents into the virtual world, it will facilitate social-presence in the following aspects: 1) Augment the social richness by extending it with being perceived as personal and intimate. Intelligent learning NPCs who act friendly and are able to provide personal services to players will offer a more social enriched environment. 2) Increase the social realism by simulating real-world people and events. Simple rule-based agents could act as door-keepers or strangers that real people may not be willing to play. Furthermore, intelligent learning NPCs could even simulate the real-life activities such as going together

to attend some events. 3) Enable players feeling of co-presence that they are together with other agents.

Perceived Interactivity: Interactivity is a key feature of many new media technologies. In a recent review of the interactivity literature, Lee, Park, and Jin [94] defined interactivity in media user terms, as the perceived degree that a person in a communication process with at least one more intelligent being can bring a reciprocal effect to other participants of the communication process by turn taking, feedback, and choice behaviors.

Perceived Usefulness: Perceived usefulness is defined as “the degree to which a person believes that using a particular system would enhance his or her job performance”. In recent years, the significant progress in building agents leads to a broader range of applications for health, military training, education in social context and so on. Intelligent agents could play roles such as animated virtual nurses for educating and counseling hospital patients and platoon sergeant who acts as a coach for trainee in mission rehearsal exercise.

3D Virtual World Skill: 3D virtual world skill often refers to the learned ability to carry out a task with pre-determined results often within a given amount of time, energy, or both in 3D virtual world. A well designed intelligent agent should be able to help the players to understand the world better and moreover, increase the 3D virtual world skill.

Enjoyment: Perceived enjoyment is adapted from Davis [95] and defined as the extent to which the activity of using a specific system is perceived to be enjoyable in it’s own right, aside from any performance consequences resulting from system use. We believe that by employing the intelligent agent, the world would be more enjoyable than the world without.

Flow: Flow describes a mental state into which individuals can slip when performing tasks with just the right amount of challenge that extend just the right amount of skill. In this state, people feel energized, focused and immersed [96]. However, unlike in video games, virtual worlds may not always have specific rules and goals to challenge and achieve. Specifically, players sometimes dont know what to do in a virtual world or may easily feel boring of it. Considering this, employing intelligent agents in the virtual world is a relatively simple solution to help improving the flow. The intelligent agents may

help players with providing information of the virtual world itself or the events held in it. Moreover, intelligent learning agents can even learn the players model and help the player explore the unknown aspect of world that he/she may be interested in accordingly.

Behavioral Intention: Behavioral intention usually indicates whether a visitor to a program or facility will return. We believe that if the agents are useful and able to help the players, player's behavioral intention of coming back to the virtual world would increase.

7.4 Evaluative Experiments

7.4.1 Research Methodology

We developed three versions of NTU Co-Space, each with a distinct type of virtual agents in the form of Non-Player Characters (NPCs). As shown in Figure 7.4, the first environment (E1) provides the baseline control condition, wherein the NPCs are only able to display static messages but do not have the capability to interact with the users. As indicate in Figure 7.5, the second environment (E2) is the first treatment condition, wherein the virtual humans are designed as embodied conversational agents using the Artificial Intelligence Mark-Up Language (AIML) [16]. AIML is an XML-compliant language that was developed by the Alicebot free software community. It is considered as a rule based repository of knowledge where the engine performs pattern matching to select the most appropriate output based on the input utterance. We have encoded as many AIML patterns as possible to enhance the conversational abilities of the agents. The third environment (E3) is the second treatment condition, wherein autonomous agents using our proposed fusion ART-based agent model are populated(as shown in Figure 7.6). Although the agents we described in the previous sections have different personalities, for the purpose of this study, we remove the variation in personality by deploying only friendly agents.

Subjects were randomly assigned to one of the three conditions. The scenario given to the subjects is that they were looking for an overseas university for an exchange program and were visiting NTU Co-Space to help them in making the decision. Each subject was asked to complete a quest in the form of a mini-game, where they would experience

Figure 7.4: A screenshot of the static NPC in E1.

Figure 7.5: A screenshot of the AIML based NPC in E2.

the key places of the NTU campus through the quest. The quest involves finding five check-points on campus where the clue to each check-point was given at the previous check-point. To maintain information equivalence across the three environments, an interactive navigation map, containing essential information of the campus, is provided to the players in all three conditions.

Figure 7.6: A screenshot of the fusion ART-based agent in E3.

The objectives of the experiment are two-fold. First, we want to observe whether deploying virtual humans in the virtual world will benefit the player's experience. Second, we assess how virtual humans with different levels of intelligence may affect the player's experience, especially in terms of the following constructs, namely *Telepresence* (TP), *Social Presence* (SP), *Perceived Interactivity* (PI), *Perceived 3D virtual world skill* (VWS), *Perceived Usefulness* (PU), *Flow* (FLW), *Enjoyment* (ENJ) and *Behavioral Intention* (to return to NTU Co-Space) (BI).

Research procedures: Subjects participated in the experiment in a computer lab. They were provided with a set of detailed instructions on the experimental procedures. Before the experiment began, the experimenter conducted a short tutorial session to familiarize the subjects with the basics of the environment and how to perform basic functions such as walking, talking and using maps in NTU Co-Space. After the tutorial, the subjects were asked to fill out a pre-questionnaire and then carry out the experiment by completing the quest given to them. For subjects in the E1 (control) condition, they completed the quest by using the map in the system to navigate the virtual world, check up information on different parts of campus and teleport to the respective checkpoints without receiving any help from NPCs. For subjects in the E2 (i.e., first treatment) condition, they were not only provided with the interactive map in the E1 condition, but they could also talk to the embodied conversational agents to ask for assistance before

teleporting through the interactive map. For subjects in the E3 (i.e., second treatment) condition, in addition to being provided with the interactive map, they were also offered the assistance of fusion ART-based NPCs that have the ability of performing autonomous behaviors both in proactive and responsive ways; moreover, since these NPCs are embedded with a Natural Language Processing module, they can understand input sentences in a flexible way. Hence, subjects were able to interact with the intelligent autonomous NPCs to request for and obtain the information they needed. Because the NPCs are autonomous, they could even offer teleport service to the specific locations requested. After the subjects have completed the quest, they filled out a post-questionnaire which assessed their experience. This questionnaire captured various aspects of user experience which would be discussed in the latter part of the paper.

Subjects: Subjects were recruited from an Introduction to Management Information Systems (MIS) class at a large Midwestern US university. In line with the institutional board review requirements, participation in this study is voluntary and will be compensated with extra credit for the course. Altogether, 58 subjects participated in this study. Specifically, 18 subjects were randomly assigned to E1, 20 subjects to E2 and 20 subjects to E3.

Pre-Questionnaire: Before the subjects started playing the *Amazing Quest*, a pre-questionnaire was used to assess the players' profile, covering demographics information, computer experience (internet, gaming, 3D virtual world), immersive tendencies, and 3D virtual world skills. Except for demographics information, all factors were measured using a five point Likert scale, from 1 (not at all) to 5 (to a very great extent).

Post-Questionnaire: After the subjects finished playing the game, another questionnaire was issued to them. The subjects were first asked for the acronym followed by the full name of the university to assess their level of recall. The main part of the questionnaire then captured the subjects' assessment of the seven constructs (as described earlier) related to NPC functions. In addition, 3D virtual world skills was also captured to examine their perceived improvement of skills after experiencing NTU Co-Space.

Among the various constructs, *Perceived Usefulness* could be objectively assessed through the time taken to complete the *Amazing Quest*. The less time spent to complete the *Amazing Quest*, the more useful the agents are. *Flow* was captured through providing the description of *Flow*, and then asking subjects to rate the degree of *Flow* they

experienced and the frequency in which they experienced *Flow*. The other constructs were captured using measurement items. At least five items were used to measure each construct. The items used to measure *Telepresence*, *Enjoyment* and *Behavioral Intention* were derived from Nah [97]. The scale for measuring *Social Presence* and *Interactivity* were adopted from Animesh et al [98]. The items were captured using a seven point Likert scale where 1 refers to strongly disagree and 7 refers to strongly agree. In order to ensure content validity of the scales used, the various measures selected for the constructs were adapted from prior studies to the context of virtual worlds as shown in Appendix A.

7.4.2 Data Analysis

Overall performance: Table 7.6 shows the overall performance of the three environments based on whether subjects completed the quest, the time taken to complete the quest and whether the subjects could recall the acronym of the virtual campus. We observe that subjects in all the three environments were able to complete the quest successfully. However, subjects in E3 spent the least amount of time and the percentage of subjects who could correctly recall the acronym of the campus is higher than those of the other two environments. This indicates that the autonomous agents deployed in E3 are more useful in helping the subjects than those of the other two environments.

Table 7.6: The overall performance of the three environments.

Evaluation Measures	E1	E2	E3
% of players complete the quest	100%	100%	100%
Time to complete the quest	20m31s	25m32s	16m56s
% of Players recall the acronym of the campus	44%	25%	45%

3D virtual world skill: Table 7.7 shows the 3D virtual world skill before and after subjects experienced NTU Co-Space in the three environments, where SD means STANDARD DEVIATION and CI represents CONFIDENCE INTERVAL. It was captured using a seven point Likert scale, where 1 refers to strongly disagree and 7 refers to strongly agree.

Table 7.7: Measurement of the 3D virtual world skill under the three environments.

Environment	Measurement	Before	After	Difference
E1	Mean	3.61	4.33	0.72
	SD	0.54	0.55	0.01
	CI	3.08-4.14	3.79-4.87	
E2	Mean	3.93	5.03	1.10
	SD	0.43	0.14	-0.29
	CI	3.51-4.34	4.88-5.17	
E3	Mean	3.50	4.66	1.16
	SD	0.60	0.33	-0.27
	CI	2.92-4.08	4.33-4.99	

Table 7.7 indicates that in E2, subjects perceived highest levels of 3D virtual world skill both before and after experiencing NTU Co-Space. However, in E3, the subjects' skill have been increased the most compared with the other two conditions. The results on confidence intervals suggest that with a confidence level of 95%, E2 has the highest rating of skills before experiencing NTU Co-Space while that of E3 is the lowest. After experiencing NTU Co-Space, subjects in E2 still have the highest rating of skills whereas subjects in E3 perceived higher skill than those in the control condition (E1). This indicates that with the interaction with autonomous agents, users feel more comfortable and perceived themselves to be more skillful with the virtual worlds.

Descriptive Statistics: Table 7.8 shows the means, standard deviations (SD), and confidence intervals (CI, with a confidence level of 95%) of ratings in E1, E2, and E3 in terms of *Telepresence* (TP), *Social Presence* (SP), *Perceived Interactivity* (PI), *Enjoyment* (ENJ), *Flow* (FLW) and *Behavior Intention* (BI). All of the constructs were measured using the seven point Likert scale. From the table, we observe that for E3, the rating of *Telepresence*, *Social Presence*, *Perceived Interactivity*, *Flow* and *Behavior Intention* have better results than E1 and E2. This means by employing autonomous agents, these factors are perceived to be stronger than those in the environment with dummy and AIML based agents. However, we note that for enjoyment, the rating in E2 is the best of the three environments. Referring to Table 7.7, subjects in E2 appear to have the best 3D virtual world skills compared to the other two. As prior work have shown that a higher level of skill is likely to enhance the feeling of enjoyment [99], we believe the rating obtained in E2 could be affected by the higher level of 3D virtual world skills.

Table 7.8: Descriptive Statistics of the six constructs measured in the three environments.

Constructs	E1			E2			E3		
	Mean	SD	CI	Mean	SD	CI	Mean	SD	CI
TP	3.86	0.34	3.64-4.08	4.08	0.49	3.76-4.40	4.41	0.42	4.14-4.68
SP	3.63	0.44	3.28-3.98	3.82	0.43	3.47-4.16	3.84	0.65	3.32-4.36
PI	4.66	0.67	4.13-5.19	4.37	0.58	3.90-4.83	5.02	0.66	4.49-5.54
ENJ	4.42	0.44	4.04-4.81	4.50	0.33	4.11-4.90	4.31	0.72	3.68-4.94
FLW	4.47	0.54	4.00-4.94	4.21	0.28	3.96-4.46	4.58	0.68	3.98-5.18
BI	4.23	0.71	3.61-4.56	4.51	0.30	4.24-4.78	4.58	0.40	4.23-4.93

Furthermore, a one-way analysis of variance (ANOVA) is used to analyze the results. Specifically, the F-test is used to evaluate the hypothesis of whether there are significant differences among the statistic data means for those constructs. The F values are calculated by the variances between conditions divided by the variance within the conditions. The p values, on the other hand, represent the probability of test statistic being different from the expected values and are directly derived from the F test. A small p value thus indicates a high confidence that the values of those constructs are different. A summary of the F values and p values among E1, E2 and E3, between E1 and E3, and between E2 and E3 are given in Table 7.9.

Table 7.9: F-test results of the six constructs under three types of comparison.

Constructs	E1, E2 & E3		E1 & E3		E2 & E3	
	F	p	F	p	F	p
TP	5.47	0.004	11.22	0.001	4.55	0.034
SP	0.58	0.561	1.10	0.295	0.01	0.903
PI	5.71	0.004	4.00	0.047	12.12	0.001
ENJ	0.15	0.862	0.33	0.566	0.17	0.68
FLW	1.62	0.199	0.28	0.595	3.12	0.079
BI	1.23	0.294	2.40	0.120	0.10	0.751

This data analysis revealed the significant effects of the three kinds of virtual humans in virtual worlds on *Telepresence* and *Perceived Interactivity*: $F(2, 519) = 5.47, p < 0.01$ for *Telepresence* and $F(2, 345) = 5.71, p < 0.01$ for *Perceived Interactivity*, where the two parameters enclosed in parentheses after F indicate the degrees of freedom of the variances between and within conditions respectively. Consistent with the statistics in Table 7.8, the fusion ART-based virtual human generate higher levels of *Telepresence* and *Perceived Interactivity* than the other two types of virtual humans, with a mean of

4.41 for E3 (versus 3.86 for E1 and 4.08 for E2) for *Telepresence*, and a mean of 5.02 for E3 (versus 4.66 for E1 and 4.37 for E2) for *Perceived Interactivity*. Although the effect of E1, E2 and E3 on *Flow* is smaller than that of *Telepresence* and *Perceived Interactivity*, the difference in *Flow* is perceived to be marginally significant between E2 and E3, with $F(1, 198) = 3.12, p < 0.1$, and a mean of 4.58 for E3 (versus 4.21 for E2). This means the *Flow* experience perceived by subjects who interacted with the fusion ART-based virtual human is stronger than those who interacted with the AIML based virtual humans. No significant difference was found for the rest of the constructs.

7.5 Summary

This chapter has reported a systematic empirical work on user study to assess whether the use of intelligent agents can improve user experience in the virtual world. By comparing the scores of the seven constructs under three environments configurations, the UX of virtual environments under each of the agent configuration is shown clearly. The result suggests that our user study has so far mostly supported the validity of our agent systems.

Chapter 8

Conclusions and Future Work

8.1 Summary of Contributions

The preceding chapters have presented several works which are aimed at addressing some of the key challenges towards building human-like agents in virtual world. The key contributions are highlighted as follows.

8.1.1 Framework for Cognitive Agents

We formalize the three key requirements for developing human-like agents in virtual environment and provide a formal definition of the human-like agent model by extending the traditional Markov Decision Process (MDP) for reinforcement learning to incorporate these requirements. The cognitive agent model incorporates explicit notions of Belief, Desire and Personality, which conjunctively drive the agents' reasoning and decision making process. Hence, it provides a unified framework for integrating the three key requirements.

8.1.2 Reinforcement Learning in Virtual Worlds

We set off to improve the interactivity and playability of virtual world and make the environment more enjoyable by employing agent technologies. We have focused on the issues and challenges for building learning agents in virtual world. To this end, we presented TD-FALCON, which enables an autonomous agent to adapt and function in a dynamic environment, as the model for implementing learning agents for personal services in virtual worlds. Given its learning capability and adaptability, it is considered

as a method towards building autonomous virtual human which is an important step towards modeling human-like agents in virtual environment.

8.1.3 Learning Personal Agents with Adaptive Player

The main objective of our work is to make the players' experience in virtual worlds more intense and personalized by employing learning player model technologies. We have focused on the capacity of agents to adapt over time to the habits and eccentric of a particular player in virtual world. To this end, we present a personal agent based on FALCON, which enables an autonomous agent to adapt and function in a dynamic environment, and augmented it with an adaptive player model for more personalized services in virtual worlds. Using learning player model technology would make the experience of interactions more personalized and let agents themselves exhibit more humanoid. This forms as part of the "personification" feature of modeling human-like agents in virtual environment

8.1.4 Self-Organizing Cognitive Models for Virtual Agents

For creating realistic virtual humans in the virtual world, we have proposed a cognitive agent architecture AIAIP that integrates goal-directed autonomy, natural language interaction and human-like personality. Extending from a family of self-organizing neural models, AIAIP maintains explicit mental representation of desires, personalities, self-awareness, situation awareness and user awareness.

We have built and deployed realistic agents in an interactive 3D virtual environment. With the virtual characters befriending and providing personalized context-aware services, we hope players will find virtual worlds more fun and appealing. To the best of our knowledge, this is perhaps one of the few in-depth works on building complete realistic agents in virtual worlds with autonomous behavior, natural interactivity and personification.

8.1.5 Evaluation of Agent based Virtual Human

The evaluation of cognitive agent model is always a difficult issue. As each cognitive agent model focuses on different aspects and the applications would be vary, there is

no universal methods to evaluate the models themselves. In this dissertation, we focus on evaluating UX of the virtual environments populated with agents rather than the standalone agents. By comparing the UX of the three virtual environments with different agent configurations, we are able to observe which kind of agent configurations outstands in UX. This evaluation methods is considered as a fair and information balanced method for agent evaluation and could be extended to other agent evaluations.

8.2 Future Work

Based on our current research, in the future, we plan to extend our work in the following directions:

8.2.1 Adaptive Player Modeling

- **Extending Player Models**

Our current player models mainly consider the player’s preferences. In the future, we also want to expand the player models to be more comprehensive. Specifically, a generic player cognitive space may be developed, which would capture a particular player’s thoughts, memories and activities into a multi-dimensional space. In this way, it would not simply provide the personal agent with players’ current interest, but also influence the decision making process directly when the agents interact with players. Consequently, the agent’s recommendation would be more accurate, natural and humanlike.

- **Implicit Player Model Acquisition**

In order to optimize the recommending agent’s accuracy, we clearly need to model the user appropriately, which is a very difficult task. In our current system, the features in player models are directly obtained from users’ query and agents’ recommendation. However, since in real world, a query is often extremely short. Sometimes, the user model constructed based on the keyword query is extremely poor. It is thus very interesting to study how to infer a user’s information need based on any implicit feedback information, which may naturally exist through the

interactions between users and agents and thus does not require any extra user effort. In fact, in the information retrieval domain, several studies have shown that implicit user modeling can improve retrieval accuracy [100–102]. In principle, each action of the player can potentially provide new evidence to help to infer the players’ need. Therefore, in the future, we may extend more channels of gathering players’ information, such as their gestures in virtual world, their dwelling time in one specific place and so on. As a matter of fact, this can also be beneficial to our personal agent in collecting the possible feedback to TD-FALCON.

8.2.2 Extending The Cognitive Models for Virtual Human

- **Integrating Emotions**

As presented in chapter 6, we have proposed a cognitive agent architecture that integrates goal-directed autonomy, natural language interaction and human-like personality. In future, we wish to develop more complete agent model that integrate emotion and facial expressions as well. From the cognition aspect, many psychology theories have proved that there is a strong relationship between emotion and cognition. The way people respond to the external events is closely tied to their implications for ones plans and goals [103]. Even some mental episodes can evoke strong emotions. From the behavior aspect, in real world, people’s nonverbal behaviors, such as facial expression, body gestures are consistent with their emotional state. The behaviors usually suggest the information about the person’s emotional arousal, personality, attitude and so on. Therefore, when creating virtual humans, it’s important to make sure the virtual character’s performance are correspond to their internal emotional state to enhance the believability.

- **Natural Language Processing**

In our work, Natural Language Processing is based on traditional approach. Moving forward, we wish to explore further into it especially the dialogue system. As nowadays, more virtual characters have taken the responsibility to help people with their daily lives. Hence, how to manage the long-term interactions is a challenge issue. There are many issues posed for long-term interactions, one of the most

critical issues is that at different phase of the interactions, the dialogue interaction would be different even with the same topic. One example would be for virtual nurses who take care of the elderly's lives would promote healthy lifestyle. At early stage when the elderly resists the recommendations and at later stage when the elderly is getting used to the living style, the way of persuasion would be quite different.

There have been some work study how to maintain long-term human agent interaction [44, 104], but few of them have studied how to manage the dialogue interaction during the long run. We have looked into the psychology theories in different domain of dialogue. We wish to propose an adaptive computational dialogue model based on these theories.

8.2.3 Extending The Evaluation of Agent Based Virtual Human

In our current user study, we have evaluated the effects of virtual humans on user experience with different constructs. Moving forward, we wish to look further into the relationship between these constructs. We have proposed a model which may suggests how different level of intelligent NPCs in 3D virtual world influences users' satisfaction. As shown in Figure 8.1, we propose that the NPCs would directly influence Telepresence and Social presence. These two constructs will impact on both Flow and Usefulness. Flow will then influence Usefulness and both of them will lead to User Satisfaction. We wish our future user study could prove this research model. Another direction of evaluation of agent based virtual human is to study how agent with different personalities affects user experience.

Figure 8.1: Proposed theoretical model and research hypotheses.

Appendix A

References for Measurement Scales

Table A.1: References for Measurement Scales

Item	Description
	Telepresence
TP1	During the virtual world tour, I felt that I was in the world the computer created.
TP2	I forgot about my immediate surroundings when I was navigating in the virtual world.
TP3	When the virtual tour ended, I felt like I came back to the "real world" after a journey.
TP4	During the virtual world tour, I forgot that I was in the middle of an experiment
TP5	The computer-generated world seemed to be "somewhere I visited" rather than "something I saw."
	Social Presence
SP1	During the virtual tour, the interaction with characters in the virtual world was personal
SP2	During the virtual tour, the interaction with characters in the virtual world was warm.
SP3	During the virtual tour, the interaction with characters in the virtual world was close.
SP4	During the virtual tour, the interaction with characters in the virtual world was humanizing.
SP5	During the virtual tour, the interaction with characters in the virtual world was meaningful.
	Interactivity
PI1	The virtual world was able to respond to my requests.
PI2	My experience during the virtual tour was interactive.
PI3	The virtual world was able to respond to my input.
PI4	The virtual world provided an interactive experience.
	3D Virtual World Skill
VWS1	I felt competent about using the 3-D virtual world environment.
VWS2	I consider myself to be experienced in the 3-D virtual world environment.
VWS3	I was skilled in carrying out various activities in the 3-D virtual world environment.
VWS4	I know a lot about activities in the 3-D virtual world environment.

Table A.2: References for Measurement Scales cont.

Item	Description
	Usefulness
USF1	I found the virtual world useful in finding information about the campus.
USF2	Using the virtual world enhanced my effectiveness in exploring the campus
USF3	I gained a better understanding of the campus through the virtual world.
USF4	The virtual world enabled me to increase my knowledge about the campus.
USF5	Using the virtual world enabled me to learn about the university more quickly.
	Enjoyment
ENJ1	I found the virtual tour to be enjoyable.
ENJ2	I found the virtual tour to be interesting.
ENJ3	I found the virtual tour to be exciting.
ENJ4	I found the virtual tour to be fun.
	Behavior Intention
BI1	I am interested to return to this virtual world site to find information about NTU when the need arises.
BI2	I would return to this virtual world site for information about NTU.
BI3	I would consider visiting this virtual world site again.
BI4	I would be interested to return to this virtual world site in the near future.
	Flow
FLW1	I experienced flow during the quest.
FLW2	When carrying out the quest, I experienced flow at some point.
FLW3	I felt I was in flow during some parts of the quest.
FLW4	I would say I was in flow during the quest.

Appendix B

Results of Principle Component Analysis

Correlation Matrix														
		TP1	TP2	TP3	TP4	SP1	SP2	SP3	SP4	SP5	PI1	PI2	PI3	PI4
Correlation	TP1	1.000	.731	.669	.763	.383	.482	.480	.235	.340	.233	.351	.205	.433
	TP2	.731	1.000	.668	.767	.302	.403	.468	.349	.382	.392	.386	.430	.473
	TP3	.669	.668	1.000	.809	.386	.392	.391	.324	.339	.379	.478	.387	.359
	TP4	.763	.767	.809	1.000	.281	.323	.312	.217	.257	.158	.275	.035	.352
	SP1	.383	.302	.386	.281	1.000	.747	.799	.777	.777	.227	.253	.129	.348
	SP2	.482	.403	.392	.323	.747	1.000	.868	.795	.895	.368	.383	.294	.427
	SP3	.480	.468	.391	.312	.799	.868	1.000	.798	.869	.396	.370	.381	.466
	SP4	.235	.349	.324	.217	.777	.795	.798	1.000	.876	.287	.355	.218	.376
	SP5	.340	.382	.339	.257	.777	.895	.869	.876	1.000	.303	.350	.262	.352
	PI1	.233	.392	.379	.158	.227	.368	.396	.287	.303	1.000	.654	.805	.739
PI2	.351	.386	.478	.275	.253	.383	.370	.355	.350	.654	1.000	.691	.869	
PI3	.205	.430	.387	.035	.129	.294	.381	.218	.262	.805	.691	1.000	.751	
PI4	.433	.473	.359	.352	.348	.427	.466	.376	.352	.739	.869	.751	1.000	

Figure B.1: Factor analysis of constructs

Correlation Matrix														
		USF1	USF2	USF3	USF4	USF5	VWS1	VWS2	VWS3	VWS4	ENJ1	ENJ2	ENJ3	ENJ4
Correlation	USF1	1.000	.647	.616	.697	.629	.207	.231	.275	.215	.283	.272	.246	.202
	USF2	.647	1.000	.815	.744	.807	.563	.490	.489	.404	.390	.316	.226	.235
	USF3	.616	.815	1.000	.802	.910	.501	.373	.471	.355	.352	.308	.178	.166
	USF4	.697	.744	.802	1.000	.769	.352	.386	.428	.441	.285	.225	.131	.155
	USF5	.629	.807	.910	.769	1.000	.514	.337	.436	.327	.283	.229	.064	.139
	VWS1	.207	.563	.501	.352	.514	1.000	.689	.821	.648	.314	.217	.067	.244
	VWS2	.231	.490	.373	.386	.337	.689	1.000	.775	.851	.325	.200	.287	.389
	VWS3	.275	.489	.471	.428	.436	.821	.775	1.000	.757	.222	.104	.073	.275
	VWS4	.215	.404	.355	.441	.327	.648	.851	.757	1.000	.172	.060	.130	.203
	ENJ1	.283	.390	.352	.285	.283	.314	.325	.222	.172	1.000	.900	.725	.808
	ENJ2	.272	.316	.308	.225	.229	.217	.200	.104	.060	.900	1.000	.778	.713
	ENJ3	.246	.226	.178	.131	.064	.067	.287	.073	.130	.725	.778	1.000	.731
	ENJ4	.202	.235	.166	.155	.139	.244	.389	.275	.203	.808	.713	.731	1.000

Figure B.2: Factor analysis of constructs cont.

Correlation Matrix															
	EoU1	EoU2	EoU3	EoU4	US1	US2	US3	US4	US5	FLW1	FLW2	FLW3	FLW4	FLW5	
Correlation	EoU1	1.000	.795	.855	.800	.335	.322	.422	.151	.227	.054	.270	.207	.015	-.055
	EoU2	.795	1.000	.852	.737	.412	.485	.545	.342	.345	.033	.298	.218	.034	-.006
	EoU3	.855	.852	1.000	.871	.478	.494	.527	.321	.343	.144	.351	.283	.090	.055
	EoU5	.800	.737	.871	1.000	.525	.545	.602	.352	.402	.222	.444	.381	.158	.088
	US1	.335	.412	.478	.525	1.000	.819	.791	.642	.662	.369	.386	.401	.203	.321
	US2	.322	.485	.494	.545	.819	1.000	.769	.751	.744	.465	.507	.442	.286	.443
	US3	.422	.545	.527	.602	.791	.769	1.000	.747	.678	.406	.507	.456	.287	.376
	US4	.151	.342	.321	.352	.642	.751	.747	1.000	.823	.517	.523	.489	.379	.531
	US5	.227	.345	.343	.402	.662	.744	.678	.823	1.000	.464	.539	.480	.363	.478
	FLW1	.054	.033	.144	.222	.369	.465	.406	.517	.464	1.000	.812	.839	.795	.731
	FLW2	.270	.298	.351	.444	.386	.507	.523	.539	.812	1.000	.886	.886	.735	.614
	FLW3	.207	.218	.283	.381	.401	.442	.456	.489	.480	.839	.886	1.000	.748	.626
	FLW4	.015	.034	.090	.158	.203	.286	.287	.379	.363	.795	.735	.748	1.000	.607
	FLW5	-.055	-.006	.055	.088	.321	.443	.376	.531	.478	.731	.614	.626	.607	1.000

Figure B.3: Factor analysis of constructs cont.

	Correlation							
	BE1	BE2	BE3	BE4	BI1	BI2	BI3	BI4
BE1	1.000	.718	.731	.660	.419	.231	.380	.386
BE2	.718	1.000	.655	.740	.566	.390	.464	.520
BE3	.731	.655	1.000	.805	.538	.372	.523	.546
BE4	.660	.740	.805	1.000	.532	.354	.452	.529
BI1	.419	.566	.538	.532	1.000	.612	.834	.883
BI2	.231	.390	.372	.354	.612	1.000	.668	.605
BI3	.380	.464	.523	.452	.834	.668	1.000	.936
BI4	.386	.520	.546	.529	.883	.605	.936	1.000

Figure B.4: Factor analysis of constructs cont.

List of Author's Publications

- [**KangTan10a**] Yi-Lin Kang and Ah-Hwee Tan. “Self-Organizing Agents for Reinforcement Learning in Virtual Worlds”, In *International Joint Conference on Neural Networks*, pages 3641–3648, Spain, 2010.
- [**KangTan10b**] Yi-Lin Kang and Ah-Hwee Tan. “Learning Personal Agents with Adaptive Player Modeling in Virtual Worlds”, In *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2010)*, pages 173–180, Toronto, Canada, 2010.
- [**TanKang10c**] Ah-Hwee Tan and Yi-Lin Kang. “Agent-Augmented Co-Space: Toward Merging of Real World and Cyberspace”, In *ATC 2010, LNCS 6407*, pages 298–312, 2010.
- [**Kang10d**] Zin-Yan Chua, Yi-Lin Kang, Xing Jiang, Kah-Hoe Pang, A. C. Gregory, Chi-Yun Tan, Wai-Lun Wong, Ah-Hwee Tan, Yew-Soon Ong, and Chun-Yan Miao. “Youth Olympic Village Co-Space”, In *International Conference on Cyberworlds*, pages 200–207, Singapore, 2010.
- [**Kang10e**] Zheng-Xiang Pan, Liang Feng, Yew-Soon Ong, Yi-Lin Kang, Ah-Hwee Tan, and Chun-Yan Miao. “Meme Selection, Variation, and Transmission in Multi-agent System”, In *World Automation Congress*, pages 1–6, Japan, 2010.
- [**Kang12a**] Yi-Lin Kang, Ah-Hwee Tan, and Fui-Hoon Nah. “Agent-based Virtual Humans in Co-Space: An Evaluative Study”, In *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2012)*, Macau, China, December 2012.

- [**Kang12b**] Yi-Lin Kang, Fui-Hoon Nah, and Ah-Hwee Tan. “Investigating Intelligent Agents in a 3D Virtual World”, In *Proceedings of International Conference on Information Systems (ICIS 2012)*, Orlando, Florida, December 2012.
- [**Kang12c**] Yi-Lin Kang, Budhitama Subagdja, Ah-Hwee Tan, Yew-Soon Ong, and Chun-Yan Miao. “Virtual Characters in Agent-Augmented Co-Space”, In *Proceedings of Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1465–1466, Valencia, Spain, June 2012.
- [**KangTan13**] Yilin Kang and Ah-Hwee Tan. “Self-organizing Cognitive Models for Virtual Agents”, In *Intelligent Virtual Agents, of 8018 Lecture Notes in Computer Science*, pages 29–43. Springer, 2013.
- [**KangTan14**] Yilin Kang and Ah-Hwee Tan. “A Self-Organizing Cognitive Agent Model for Virtual Human”, In submission to *Journal of Autonomous Agents and Multi-Agent Systems*

References

- [1] A.-H. Tan, “Self-organizing Neural Architecture for Reinforcement Learning,” in *International Symposium on Neural Networks*, pp. 470–475, 2006.
- [2] A.-H. Tan, G. A. Carpenter, and S. Grossberg, “Intelligence through Interaction: Towards a Unified Theory for Learning,” in *Proceedings of International Symposium on Neural Networks*, pp. 1094–1103, 2007.
- [3] D. Thomas and J. Brown, “Why virtual worlds can matter,” *International Journal of Media and Learning*, vol. 4, pp. 37–49, 2009.
- [4] J. Weinstein and J. Myers, “Same principles apply to virtual world expansion as to china and other new markets,” *Media Village*, vol. 11, 2006.
- [5] D. Jan, A. Roque, A. Leuski, J. Morie, and D. Traum, “A virtual tour guide for virtual worlds,” in *Intelligent Virtual Agents Conference (IVA)*, pp. 372–378, 2009.
- [6] D. Friedman, A. Steed, and M. Slater, “Spatial social behavior in second life,” *Lecture Notes in Computer Science*, vol. 4722, pp. 252–263, 2007.
- [7] P. Doyle, “Virtual intelligence from articial reality: Building stupid agents in smart environments,” in *AAAI Spring Symposium on Artificial Intelligence and Computer Games*, pp. 972–978, 1999.
- [8] I. Umarov and M. Mozgovoy, “Believable and effective ai agents in virtual worlds: Current state and future perspectives,” *International Journal of Gaming and Computer-Mediated Simulations*, vol. 4, pp. 37–59, 2012.

- [9] J. Gratch, J. Rickel, E. Andr, J. Cassell, and E. Petajan, “Creating interactive virtual humans: Some assembly required,” *IEEE Intelligent Systems*, vol. 14, pp. 54–63, 2002.
- [10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An Introduction*. MIT Press, 1998.
- [11] I. Millington, *Artificial intelligence for games*. Morgan Kaufmann, 2006.
- [12] D. Wang, B. Subagdja, A.-H. Tan, and G.-W. Ng, “Creating human-like autonomous players in real-time first person shooter computer games,” in *proceedings, Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence (IAAI’09)*, pp. 173–178, 2009.
- [13] R. Aylett and M. Luck, “Applying artificial intelligence to virtual reality: Intelligent virtual environments,” *Applied Artificial Intelligence*, vol. 14(1), pp. 3–32, 2000.
- [14] A.-H. Tan, N. Lu, and D. Xiao, “Integrating Temporal Difference Methods and Self-Organizing Neural Networks for Reinforcement Learning with Delayed Evaluative Feedback,” *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 230–244, 2008.
- [15] G. A. Carpenter and S. Grossberg, “Adaptive Resonance Theory,” in *The Handbook of Brain Theory and neural Networks*, pp. 87–90, MIT Press, 2003.
- [16] R. S. Wallace, “The anatomy of A.L.I.C.E.,” *Tech.report, ALICE AI Foundation*, 2000.
- [17] D. M. Bourg and G. Seemann, *AI for Game Developers*. O’Reilly Media, 2004.
- [18] D. Rousseau and B. Hayes-Roth, “A social-psychological model for synthetic actors,” *Stanford Knowledge Systems Laboratory Report KSL-97-07*, 1997.
- [19] J. Weizenbaum, “ELIZA: a computer program for the study of natural language communication between men and machines,” *Communications of the ACM*, vol. 9, 1996.

- [20] S. Kopp, L. Gesellensetter, N. C. Krmer, and I. Wachsmuth, “A conversational agent as museum guide - design and evaluation of a real-world application,” *Intelligent Virtual Agents*, pp. 329–343, 2005.
- [21] P. B.-D. Byl, *Programming Believable Characters for Computer Games*. Charles River Media, 2004.
- [22] T. S. Trias, S. Chopra, B. D. Reich, M. B. Moore, N. I. Badler, B. L. Webber, and C. W. Geib, “Decision networks for integrating the behaviors of virtual agents and avatars,” in *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS '96)*, 156–162. Los Alamitos, CA: IEEE Computer, pp. 156–162, 1996.
- [23] D. Allison, B. Wills, D. Bowman, J. Wineman, and L. F. Hodges, “The virtual reality gorilla exhibit,” *IEEE Comput. Graph. Appl.*, vol. 17, pp. 30–38, Nov. 1997.
- [24] J. L. Asbahr, “Beyond: A portable virtual world simulation framework,” in *Proc. of 7th Intl Python Conf*, 1998.
- [25] S. Chopra-Khullar and N. I. Badler, “Where to look? automating attending behaviors of virtual human characters,” in *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, pp. 16–23, 1999.
- [26] K. Perlin and A. Goldberg, “Improv: A system for scripting interactive actors in virtual worlds,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 205–216, 1996.
- [27] D. Xu and H. Wang, “Intelligent agent supported personalization for virtual learning environments,” *Decision Support Systems*, vol. 42(2), pp. 825–843, 2005.
- [28] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition ed., 2003.
- [29] S. W. McQuiggan and J. C. Lester, “Modeling and evaluating empathy in embodied companion agents,” *Int. J. Hum.-Comput. Stud.*, vol. 65, no. 4, pp. 348–360, 2007.

- [30] M. Looks, B. Goertzel, and C. Pennachin, “Novamente: An integrative architecture for general intelligence,” in *Proceedings of AAAI Symposium on Achieving Human-Level Intelligence through Integrated Systems and Research*, pp. 54–61, 2004.
- [31] S. Rabin, *Introduction to Bayesian Networks and Reasoning Under Uncertainty*. Charles River Media, 2002.
- [32] C. Sierra, C. Castelfranchi, K. S. Decker, and J. S. Sichman, eds., *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 1*, IFAAMAS, 2009.
- [33] S. Yoon, R. Burke, B. Blumberg, and G. Schneider, “Interactive training for synthetic characters,” in *AAAI*, pp. 249–254, 2000.
- [34] M. Gerhard, D. Moore, and D. Hobbs, “Embodiment and copresence in collaborative interfaces,” *Int. J. Hum.-Comput. Stud.*, vol. 64(4), pp. 453–480, 2004.
- [35] A. Leuski and D. Traum, “A statistical approach for text processing in virtual humans,” in *26th Army Science Conference*, 2008.
- [36] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen, “Interactive storytelling: A player modelling approach,” in *Proceedings of the third Artificial Intelligence and Interactive Digital Entertainment conference*, 2007.
- [37] M. Sharma, S. Ontan, C. Strong, M. Mehta, and A. Ram, “Towards player preference modeling for drama management in interactive stories,” in *In Proceedings of the Twentieth International FLAIRS Conference (FLAIRS07)*, AAAI Press, 2007.
- [38] D. Charles and M. Black, “Dynamic player modeling: A framework for player-centred digital games,” in *International Conference on Computer Games: Artificial Intelligence, Design and Education*, 2004.
- [39] A. J. Lockett, C. L. Chen, and R. Miikkulainen, “Evolving explicit opponent models in game playing,” in *in: Proceedings of the Genetic and Evolutionary Computation Conference*, 2007.

- [40] R.J.S.Baker, P.I.Cowling, T.W.G.Randall, and P.Jiang, “Can opponent models aid poker player evolution?,” in *IEEE Symposium on Computational Intelligence and Games*, 2008.
- [41] R. Houlette, *Player Modelling for Adaptive Games*. Charles River Media, 2004.
- [42] S. Hladky and V. Bulitko, “An evaluation of models for predicting opponent locations in first-person shooter video games,” in *IEEE Symposium on Computational Intelligence and Games*, 2008.
- [43] M. van der Heijden, S. Bakkes, and P. Spronck, “Dynamic formations in real-time strategy games,” in *IEEE Symposium on Computational Intelligence and Games*, 2008.
- [44] T. Bickmore, Rosalind, and W. Picard, “Establishing and maintaining long-term human-computer relationships,” *ACM Transactions on Computer Human Interaction*, vol. 12, pp. 293–327, 2005.
- [45] R. Gockley, J. Forlizzi, and R. Simmons, “Interactions with a moody robot,” in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, HRI ’06, (New York, NY, USA), pp. 186–193, ACM, 2006.
- [46] K. Merrick and M. Maher, “Motivated reinforcement learning for non-player characters in persistent computer game worlds,” in *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, ACE ’06, (New York, NY, USA), ACM, 2006.
- [47] P. Kenny, T. Parsons, J. Gratch, and A. Rizzo, “Evaluation of justina: A virtual patient with ptsd,” in *Proceedings of the intelligent virtual agent*, pp. 394–408, 2008.
- [48] L. Edward, D. Lourdeaux, J. Barths, D. Lenne, and J. Burkhardt, “Modelling autonomous virtual agent behaviours in a virtual environment for risk,” *International Journal of Virtual Reality*, vol. 7, pp. 13–22, 2008.

- [49] Z. Kasap, M. Moussa, P. Chaudhuri, and N. Thalmann, “Making them remember-emotional virtual characters with memory,” *IEEE Computer Graphics and Applications*, vol. 29, pp. 20–29, 2009.
- [50] H.-Q. Chong, A.-H. Tan, and G.-W. Ng, “Integrated cognitive architectures: a survey,” *Artif Intell Rev*, vol. 28, pp. 103–130, 2007.
- [51] A. Newell, *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press, 1990.
- [52] J. E. Laird, A. Newell, and P. S. Rosenbloom, “Soar: An architecture for general intelligence,” *Artificial Intelligence*, vol. 33, pp. 1–64, 1987.
- [53] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, “An integrated theory of the mind,” *Psychological Review*, vol. 111, pp. 1036–1060, 2004.
- [54] P. Langley and D. Choi, “A unified cognitive architecture for physical agents,” in *Proceedings, Twenty-First National Conference on Artificial Intelligence*, pp. 1469–1474, 2006.
- [55] M. Bratman, D. Israel, and M. Pollack, “Plans and resource-bounded practical reasoning,” *Comput Intell*, vol. 4, no. 4, p. 349355, 1988.
- [56] M. Georgeff and F. Ingrand, “Decision-making in an embedded reasoning system,” in *Proceedings, international joint conference on artificial intelligence*, pp. 972–978, 1989.
- [57] R. Brooks, *Cambrian Intelligence: The Early History of the New AI*. Boston, MA: MIT Press, 1999.
- [58] H. Nakashima and I. Noda, “Dynamic subsumption architecture for programming intelligent agents,” in *Proceedings, IEEE International Conference on Multi-Agent systems*, pp. 190–197, 1998.
- [59] D. Toal, C. Flanagan, C. Jones, and B. Strunz, “Subsumption architecture for the control of robots,” in *IMC-13*, 1996.

- [60] R. A. Brooks, “How to build complete creatures rather than isolated cognitive simulators,” in *Proceedings, Architectures for Intelligence*, pp. 225–240, 1991.
- [61] J. Bryson, A. Smaill, and G. Wiggins, “The reactive accompanist: Applying subsumption architecture to software design,” tech. rep., Department of Artificial Intelligence, University of Edinburgh, 2005.
- [62] R. Sun, “Cognitive science meets multi-agent systems: a prolegomenon,” *Philosophical Psychology*, vol. 14, no. 1, pp. 5–28, 2001.
- [63] R. Sun and T. Peterson, “Learning in reactive sequential decision tasks: The CLARION model,” in *Proceedings, IEEE International Conference on Neural Networks*, pp. 1073–1078, 1996.
- [64] R. Sun and T. Peterson, “Hybrid learning incorporating neural and symbolic processes,” in *Proceedings, IEEE International Conference on Fuzzy Systems*, pp. 727–732, 1998.
- [65] R. Sun, P. Slusarz, and C. Terry, “The interaction of the explicit and the implicit in skill learning: A dual-process approach,” *Psychological Review*, vol. 112, no. 1, pp. 159–192, 2005.
- [66] R. Sun and X. Zhang, “Accounting for a variety of reasoning data within a cognitive architecture,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 18, no. 2, pp. 169–191, 2006.
- [67] A. Bogdanovych, J. A. Rodriguez, S. Simoff, and A. Cohen, “Virtual agents and 3d virtual worlds for preserving and simulating cultures,” in *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, pp. 271–284, 2009.
- [68] W. Swartout, “Lessons learned from virtual humans,” *AI Magazine*, vol. 31, pp. 9–20, 2010.
- [69] M. Triola, H. Feldman, A. Kalet, S. Zabbar, E. K. Kachur, and C. Gillespie, “A randomized trial of teaching clinical skills using virtual and live standardized patients,” *Journal of General Internal Medicine*, vol. 21, pp. 424–429, 2006.

- [70] A. R. Chaturvedi, D. R. Dolk, and P. L. Drnevich, "Design principles for virtual worlds," *MIS Quaterly*, vol. 35, pp. 673–684, 2011.
- [71] K. L. Nowak and F. Biocca, "The effect of the agency and anthropomorphism on users' sense of telepresence, copresence, and social presence in virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 12, pp. 481–494, 2003.
- [72] W. Barfield and C. Hendrix, "The effect of update rate on the sense of presence within virtual environments," *Virtual Reality: The Journal of the Virtual Reality Society*, vol. 1, pp. 3–16, 1995.
- [73] M. Georgeff and F. Ingrand, "Decision-making in an embedded reasoning system," in *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 972–978, 1989.
- [74] Z. Kasap and N. Thalmann, "Intelligent virtual humans with autonomy and personality: State-of-the-art," *Intelligent Decision Technologies*, vol. 1, pp. 3–15, 2007.
- [75] S. Franklin and A. Graesser, "Is it an agent, or just a program?: A taxonomy for autonomous agents," in *Intelligent Agents III. Agent Theories, Architectures, and Languages*, pp. 21–35, 1997.
- [76] J. Jensen, "'interactivity' tracking a new concept in media and communication studies," *NORDICOM Review*, vol. 19, no. 1, pp. 185–205, 1998.
- [77] A.-H. Tan, "FALCON: A Fusion Architecture for Learning, COgnition, and Navigation," in *Proceedings of International Joint Conference on Neural Networks*, pp. 3297–3302, 2004.
- [78] G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System," *Neural Networks*, vol. 4, pp. 759–771, 1991.
- [79] Y.-L. Kang and A.-H. Tan, "Self-organizing agents for reinforcement learning in virtual worlds," in *to be appear in: proceedings, International Joint Conference on Neural Networks*, 2010.

- [80] A.-H. Tan and H.-S. V. Soon, “Concept Hierarchy Memory Model: A neural architecture for conceptual knowledge representation, learning, and commonsense reasoning,” *International Journal of Neural Systems*, vol. 7, no. 3, pp. 305–319, 1996.
- [81] A.-H. Tan, “Adaptive resonance associative map,” *Neural Networks*, vol. 8, pp. 437–446, 1995.
- [82] R. Mccrae and P. Costa, “An introduction to the five-factor model and its applications,” *Journal of Personality*, vol. 60, pp. 172–215, 1992.
- [83] R. Smullyan, *First-Order Logic, 2nd ed.* Dover Publications, 1968/1995.
- [84] R. Brachman, A. Borgida, D. McGuinness, P. Patel-Schneider, and L. Resnick, “The classic knowledge representation system or, kl-one: The next generation,” in *FGCS*, pp. 1036–1043, 1992.
- [85] D. Ferrucci, E. Brown, J. Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, “Building watson: An overview of the deepQA project,” *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010.
- [86] A. Ritter, S. Clark, and O. Mausam, “Named entity recognition in tweets: An experimental study,” in *Proc. Empirical Methods in Natural Language Processing*, 2011.
- [87] R. Pfeifer and C. Scheier, *Understanding Intelligence*. The MIT Press, 2001.
- [88] Unity-Technologies, “Unity 3d home page.” <http://unity3d.com>, 2011.
- [89] E. Law, V. Roto, A. Vermeeren, J. Kort, and M. Hassenzahl, “Towards a shared definition of user experience,” in *CHI08*, pp. 2395–2398, 2008.
- [90] ISO, “Ergonomics of human system interaction - part 210: Human-centered design for interactive systems,” Tech. Rep. ISO FDIS 9241-210, International Organization for Standardization, Geneva, Switzerland.

- [91] J. Nielsen and D. Norman, “The definition of user experience.” <http://www.nngroup.com/articles/definition-user-experience/>.
- [92] J. Miller, “The user experience,” *IEEE Internet Computing*, vol. 9, pp. 90–92, 2005.
- [93] M. Lombard and T. Ditton, “At the heart of it all: The concept of presence,” *Journal of Computer Mediated Communication*, vol. 3, no. 2, 1997.
- [94] K. Lee, N. Park, and S. Jin, “Narrative and interactivity in computer games,” in *Playing video games: Motives, responses, and consequences*, pp. 304–322, 2006.
- [95] F. Davis, “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” *MIS Quart*, vol. 13, no. 3, pp. 319–339, 1989.
- [96] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York: Harper & Row, 1990.
- [97] F. Nah, B. Eschenbrenner, and D. DeWester, “Enhancing brand equity through flow and telepresence: A comparison of 2d and 3d virtual worlds,” *MIS Quarterly*, vol. 35, pp. 731–748, 2011.
- [98] A. Animesh, A. Pinsonneault, S.-B. Yang, and W. Oh, “An odyssey into virtual worlds: Exploring the impacts of technological and spatial environments,” *MIS Quarterly*, vol. 35, pp. 789–810, 2011.
- [99] M.-Y. Yi and Y. Hwang, “Predicting the use of web-based information systems: self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model,” *International Journal of computer-human studies*, vol. 59, pp. 431–449, 2003.
- [100] M. Claypool, P. Le, M. Waseda, and D. Brown, “Implicit interest indicators,” in *INTELLIGENT USER INTERFACES*, pp. 33–40, ACM Press, 2001.
- [101] T. Joachims, “Optimizing search engines using clickthrough data,” in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 133–142, 2002.

- [102] K. Sugiyama, K. Hatano, and M. Yoshikawa, “Adaptive web search based on user profile constructed without any effort from users,” in *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pp. 675–684, 2004.
- [103] A. Ortony, *The Cognitive Structure of Emotions*. Cambridge University Press, 1990.
- [104] M. Riedl, C. J. Saretto, and R. M. Young, “Managing interaction between users and agents in a multi-agent storytelling environment,” in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, (New York, NY, USA), pp. 741–748, ACM, 2003.