Agent mediated interactive storytelling

Cai, Yundong

2011

Cai, Y. D. (2011). Agent mediated interactive storytelling. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/45767>

<https://doi.org/10.32657/10356/45767>

Agent Mediated Interactive Storytelling

Yundong Cai

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree
of Doctor of Philosophy

2011

ABSTRACT

Interactive storytelling in the virtual environment attracts many interests from researchers and educators in recent years. People are engaged in the storytelling as active participants by interacting with virtual characters as well as story director (storyteller), rather than acting as passive audiences in the conventional storytelling. However, it is a great challenge to create an immersive interactive storytelling system in the virtual environment from scratch with an engaging storyline and believable characters, especially for the novice developers without many programming skills. The current interactive storytelling methodologies (e.g. plot-based interactive storytelling and character-based interactive storytelling) cannot ensure the story plot coherence and character believability simultaneously. The director and characters are not autonomous and intelligent enough to handle dynamic user interactions and context changes.

In the thesis, we propose a novel **Agent-Oriented Methodology for Interactive Storytelling** (AOMIS) to facilitate novices to develop an immersive interactive storytelling system from a story. It generates both the story plot and the character behaviors as agent goals seamlessly in real-time. Human-like agent is a promising solution for the intelligent director and characters in the interactive virtual storytelling with the high-level goal automation besides the low-level behavior automation. A multi-agent system (MAS) has been constructed for illustrating the hybrid model, which involves a scriptwriter agent to draft the nonlinear storylines, a director agent to select the storyline, virtual character agents to perform the story scenes, and other supporting agents.

In contrast with other studies, Fuzzy Cognitive Goal Net (FCGN) is developed incorporating a plot model as well as a character model involving agents in the interactive storytelling. It provides high-level intelligence with goal modeling compared to action modeling used

in other interactive storytelling approaches, and it enables better interaction experience through the fuzzy cognitions over the user interactions and state of the virtual world intelligently. As the context model, Evolutionary Fuzzy Cognitive Map (E-FCM) is proposed to simulate the probabilistic and fuzzy causal relationships among the story scenes, user preferences, real-time interactions, and state of the virtual world in order to achieve a reasonable storyline.

An interactive storytelling engine, Multi-Agent Development Environment for Interactive Storytelling (S-MADE) has been implemented based on the agent-oriented interactive storytelling methodology, which enables the designers to realize an interactive storytelling scenario in the virtual environment easily and intuitively.

In order to verify the proposed methodology, case studies have been implemented in the research projects and examined in secondary schools, wherein an enhanced immersive experience for the users is observed. By using interactive storytelling as a teaching and learning method, it promotes the learning interests and efficiency of students in a positive manner. Moreover, it shows a high potential for the teachers to construct the interactive storytelling through the proposed methodology easily and effectively.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to the following people for their support, encouragement and help during my Ph.D study at Nanyang Technological University.

First and foremost, I would like to take this opportunity to pay a high tribute to my PhD supervisor **Dr. Miao Chun Yan**, for her continuous support and numerous guidance at my research work as well as life in the academic years. Dr. Miao provides great freedom for my research, and was always willing to listen and share her experiences with me. Moreover, she often discussed with me about potential research directions over time, which inspired me in the area of agent-mediated interactive storytelling.

I'd also like to thank my co-supervisor, **Dr. Tan Ah-Hwee**, who is most helpful by providing insightful comments and reviewing my papers. Moreover, special thanks go to **Dr. Shen Zhiqi**, for a lot of inspirations from him in the discussions and technical guidance.

Thanks also go to Emerging Research Lab, School of Computer Engineering, Nanyang Technological University, for the financial support and computing facilities, and helpful lab technicians **Wong Kia Yan** and **Thin Nandar Soe**.

Dr. Weng Jianshu gave me strong support, great advice and discreet work to improve my thesis. I also would like to thank other team members at developing the projects of serious games and interactive storytelling, who are **Benny Tan**, **Bai Xiaoguang**, **Yu Han**, **Guo Zhengjing**, **Zheng Yutin** and **Ji Jun**. **Benny Tan** and **Gao Shan** also helped me a lot at my thesis proofreading.

Last but not least, I would like to thank my wife **Xia**, my mum, and my sweet sons **Weiheng**, **Weibo** for their invaluable love and encouragements always.

NOMENCLATURE

API	Application Programming Interface
AOMIS	Agent-Oriented Methodology for Interactive Storytelling
AVLE	Agent-based Virtual Learning Environment
A-MUVE	Agent-based Multi-user Virtual Environment
BDI	Belief, Desire and Intention Agent Architecture
CBIS	Character-based Interactive Storytelling
CM	Cognitive Map
E-FCM	Evolutionary Fuzzy Cognitive Map
FCM	Fuzzy Cognitive Map
FCGN	Fuzzy Cognitive Goal Net
GN	Goal Net
IS	Interactive Storytelling
IVS	Interactive Virtual Storytelling
MADE	Multi-agent Development Environment
MAS	Multi-agent System
MUVE	Multi-user Virtual Environment
NPC	Non-player Character
PBIS	Plot-based Interactive Storytelling
HTN	Hierarchical Task Network
S-MADE	Multi-agent Development Environment for Interactive Storytelling
VLE	Virtual Learning Environment

CONTENTS

Abstract	i
Acknowledgments	iii
Nomenclature	iv
List of Figures	x
List of Tables	xiv
1 Introduction	1
1.1 Interactive Storytelling	1
1.2 Motivation	2
1.3 Problem Statement	3
1.4 Objectives	4
1.5 Summary of Contributions	5
1.6 Thesis Structure	6
2 Literature Review and Theoretical Background	9
2.1 Introduction	9
2.2 Story, Storytelling, Interactive Storytelling and Virtual Storytelling	9
2.2.1 Story and Storytelling	9
2.2.2 Interactive Storytelling and Virtual Storytelling	11
2.2.3 Plot-based Interactive Storytelling	12
2.2.4 Character-based Interactive Storytelling	15
2.2.5 Hybrid Interactive Storytelling	16
2.2.6 Narratology	19

2.3	Theoretical Background	19
2.3.1	Agent	19
2.3.2	Goal Net	22
2.3.3	Fuzzy Cognitive Map	24
2.4	Summary	26
3	Agent-Oriented Methodology for Interactive Storytelling (AOMIS)	27
3.1	Introduction	27
3.2	Agents and Goals in Interactive Storytelling	28
3.3	Storytelling Elements Definitions	29
3.4	Structure Model of Interactive Storytelling	31
3.4.1	Agents	31
3.4.2	Audience/User/Participant	32
3.4.3	Story Context	34
3.4.4	User Interactions	35
3.4.5	Interactive Storytelling Methodology Overview	35
3.5	Interactive Storytelling Process	38
3.5.1	Story Authoring	38
3.5.2	Story Executing/Scene Selection	40
3.5.3	Scene Dispatching	40
3.5.4	Character Performing	40
3.6	Steps to Create Interactive Storytelling	41
3.6.1	Identify the Character Agents from the Story	41
3.6.2	Identify the Goals and Actions for Each Agent and Conditions to Handle User Interactions	42
3.6.3	Design Goals and Actions with Fuzzy Cognitive Goal Net Designer	43
3.6.4	Running Characters' Goal Net with S-MADE Runtime	46
3.7	DIRACT	47
3.7.1	Structure of DIRACT Storytelling System	47
3.7.2	DIRACT Agent	48
3.8	Methodology Evaluation	49
3.9	Summary	49

4	Plot Model and Character Model: Fuzzy Cognitive Goal Net (FCGN)	51
4.1	Introduction	51
4.2	Model Description	52
4.2.1	Basic Concept of Fuzzy Cognitive Goal Net	52
4.2.2	Elements of Fuzzy Cognitive Goal Net	53
4.2.3	BDI View of Fuzzy Cognitive Goal Net	56
4.2.4	Goal Operation	56
4.2.5	Goal Evaluation	58
4.2.6	Running Algorithm	59
4.3	Plot Model of Interactive Storytelling	62
4.3.1	Fuzzy Cognition in Interactive Storytelling	63
4.4	Character Model in Interactive Storytelling	65
4.5	Hybrid Interactive Storytelling by Combining Plot Model and Character Model	66
4.6	Discussions	67
4.6.1	Comparisons to Other Models	67
4.7	Summary	69
5	Context Model: Evolutionary Fuzzy Cognitive Maps (E-FCM)	70
5.1	Introduction	70
5.2	Context Modeling in Interactive Virtual Storytelling	73
5.2.1	Context Representation	73
5.3	Game World Modeling with Evolutionary Fuzzy Cognitive Map (E-FCM) .	74
5.3.1	Concepts	76
5.3.2	Causal Relationship	78
5.3.3	Weight Adjustments with Interactive Evolutionary Computing . . .	80
5.3.4	Pseudo-code of Running E-FCM in Real-time	82
5.3.5	Game World Context Modeling Procedure with E-FCM	83
5.3.6	Comparisons to Other Computational Models	85
5.4	Case Study: Village of Mystery	86
5.4.1	Experiment Setup	87
5.4.2	Experiment Results	90
5.4.3	Evaluations	94
5.4.4	Discussions	95
5.5	Summary	96

6	Agent-based Interactive Storytelling Engine: S-MADE	97
6.1	Introduction	97
6.2	Multi-Agent Development Environment for Storytelling (S-MADE)	98
6.2.1	Background: Multi-Agent Development Environment	98
6.2.2	S-MADE Components	99
6.2.3	Fuzzy Cognitive Goal Net Designer	101
6.2.4	S-MADE Runtime for Single Agent	103
6.2.5	S-MADE Running Process	105
6.3	Evaluation	107
6.3.1	Comparisons with Other Storytelling Methodologies	108
6.3.2	Comparisons with Other Storytelling Systems	109
6.3.3	User Interaction Evaluation	109
6.4	Summary	110
7	Applications: Interactive Virtual Storytelling for Education	111
7.1	Introduction	111
7.2	Virtual Singapura	112
7.2.1	Project Background	112
7.2.2	Story Implementation	113
7.2.3	Results and Discussions	117
7.3	Chronicles of Singapura	119
7.3.1	Project Background	119
7.3.2	Agent-based Virtual Learning Environment (AVLE) with Interactive Storytelling	120
7.3.3	Meet Teacher at School Before the Exploration	122
7.3.4	Adventure in the Plant Transportation System	125
7.3.5	Assessments	131
7.4	Summary	134
8	Conclusions and Future Recommendations	135
8.1	Summary of Contributions	135
8.2	Limitations	138
8.3	Future Recommendations	138
8.3.1	Refine Fuzzy Cognitive Goal Net Model	138

8.3.2	Develop Objective Evaluation Metrics	141
8.3.3	Text to Scene or Speech to Scene	142
Appendices		143
A.1	List of Publications	143
A.2	Multi-agent Development Environment for Storytelling (S-MADE) Code . .	145
A.3	Evolutionary Fuzzy Cognitive Map (E-FCM) Code	148
A.4	Questionnaire for School Test of Chronicles of Singapura	158
References		166

LIST OF FIGURES

2.1	Six Components of Tragedy (Aristotle, 330BC)	11
2.2	Tension vs. Time at Different Story Stages	12
2.3	Storytelling Architecture of Mimesis Project	13
2.4	Hierarchical Task Network for Character Action Planning	16
2.5	Screenshot of Two Characters in Façade [48]	17
2.6	Graphical User Interface of Storytelling Alice	19
2.7	A Sample Hierarchical Goal Net	23
2.8	Goal Net Transitions for Goals G_i , G_j and G_k : (a) Sequence (b) Concurrency (c) Choice (d) Synchronization	24
2.9	A Sample Fuzzy Cognitive Map (Bart Kosko)	25
3.1	Multi-agent System View for Interactive Storytelling Hybrid System	31
3.2	A Full Picture of Agent-Based Methodology for Interactive Storytelling . .	36
3.3	Interactive Storytelling Process: Agents vs. Goals. (a) The Scriptwriter Agent Designs Story Plot with Various Choices; (b) The Director Agent Selects Story Scenes Dynamically; (c) The Story Scene S_2 Involves Perfor- mance between Two Actors; (d) The Director Agent Dispatches the Scene to Two Actors as Their Behaviors.	39
3.4	A Linear “Little Red Riding Hood” Story	44
3.5	A Non-linear “Little Red Riding Hood” Story with Alternative Goals of Director Agent	44
3.6	The “Detect the False Grandma” Scene in “Little Red Riding Hood” Story	45
3.7	Fuzzy Cognitive Map in the Scene of “Detect False Grandma”	45

3.8	Behaviors of Little Red Riding Hood in the Scene “Detect the False Grandma”	46
3.9	Multi-agent Interactive Storytelling System of DIRACT	47
4.1	A Sample Fuzzy Cognitive Goal Net	52
4.2	Transition Types of Fuzzy Cognitive Goal Net: (a) Direct; (b) Parallel; (c) Choice; (d) Synchronization	55
4.3	Goal Operation of Goals G_i and G_j (a) Union; (b) Intersection; (c) Not; (d) Following-by	57
4.4	Fuzzy Cognitive Goal Net with Alternative Storylines	62
4.5	Fuzzy Cognitive Goal Net with Alternative Behaviors of Little Red Riding Hood	65
4.6	Fuzzy Cognitive Goal Net with Plot and Behaviors of Little Red Riding Hood Story	67
5.1	Structure Overview of E-FCM (E is the System Enclosure, Clock is the Reference Clock, S_i is the Concept and R_i is the Causal Relationship) . . .	75
5.2	Flow Diagram of Weight Adjustments with Interactive Evolutionary Computing	82
5.3	Game World Context Modeling Procedure with E-FCM	84
5.4	The E-FCM Model of the Sample Story Scenario (The Part Inside the Solid Line Shows the Character Model and the Part Outside Shows the Context Model. ‘+’ Means “Positive Causal Relationship” and ‘-’ Means “Negative Causal Relationship”.)	87
5.5	Results of Concepts “Mosquito”, “Health” and “Energy”: (a) First Round Experiment of E-FCM; (b) Second Round Experiment of E-FCM; (c) Third Round Experiment of E-FCM.	91
5.6	Results After Changing Value of “Clean” to 1 Following Simulation Shown in Figure 5.5(b)	92
5.7	Running Results After Changing Time Schedule of “Health” from 1 to 5. .	93
5.8	Running Results of the Scenario with FCM	93
5.9	Screenshots at the First Round Experiment: (a) Avatar with Full Health; (b) Avatar with Decreased Health with Dirty Water and Mosquitoes Nearby	94
6.1	Overall Architecture of S-MADE	99

6.2	Interface of Fuzzy Cognitive Goal Net Designer (Left Panel: Buttons to Add Goal, Transition and Tasks; Right Panel: Drawing Panel of Fuzzy Cognitive Goal Net)	102
6.3	Units of S-MADE Runtime to Handle Single Agent	104
7.1	Fuzzy Cognitive Goal Net for Illness Investigation Scenario by the Scriptwriter Agent	114
7.2	Storyline Generated by the Director Agent (Marked in the Dark Color) . .	114
7.3	“Visit Hospital” Scene	115
7.4	Screenshots within the Hospital Scene	116
7.5	Fuzzy Cognitive Map in the Hospital Scene	117
7.6	Learning Structure in the Agent-based Virtual Learning Environment . . .	120
7.7	Goal Net for the School Scene	123
7.8	Goal Net for DIRACT Agent “Teacher”	123
7.9	Goal Net for DIRACT Agent “Student”	123
7.10	Snapshots for (a) Scene Initialization (Teacher Wandering and Students Gathering); (b) Asking the Protagonist to Explore inside the Banana Tree; and (c) Ask the Protagonist to Teach the Inner Structure of Tree	124
7.11	Fuzzy Cognitive Goal Net of Lab Supervisor Agent to Choose the Learning Content	126
7.12	Lab Supervisor “Miss Lee” Greets Students with an Introduction (‘Greet Student’ Goal in Figure 7.11)	127
7.13	Diffusion Experiment with 2D Simulation: Add Ink Drops to Observe the Movements of Molecules of Diffusion (‘Experiment’ Goal in Figure 7.11) . .	127
7.14	Fuzzy Cognitive Goal Net of Director Agent to Control the Role-playing of Students	128
7.15	Sub-goal of Director Agent to Visit Plant Transportation	128
7.16	Student is Flying Upwards inside of Stem Xylem (‘Visit Stem’ Goal in Figure 7.14)	129
7.17	Cross Structure of Leaf: Xylem on Top and Phloem at Bottom (‘Visit Leaf’ Goal in Figure 7.14)	129
7.18	The Player Pushes the Water Molecule Towards a Carbon-dioxide Molecule to Generate Food (‘Generate Food’ Goal in Figure 7.14)	130

7.19 Fuzzy Cognitive Goal Net of Water Molecule Agent to Go Through Root, Stem, Leaf and Generate the Food	130
7.20 Distribution of Scores in Agent-based Virtual Learning Environment (AVLE) and Classroom Learning (CL)	132
7.21 Average of Each Question in Agent-based Virtual Learning Environment (AVLE) and Classroom Learning (CL)	133

LIST OF TABLES

3.1	Agents and Their Goals in Interactive Storytelling	29
3.2	Comparison of Director and Character	33
3.3	Feature Comparisons with Mimesis, Character-based IS and Façade	50
4.1	Task Pool	55
5.1	Weight Matrix of the Causal Relationship	89
5.2	Conditional Probability Matrix of the Causal Relationship	89
6.1	Comparisons with Plot-based and Character-based Interactive Storytelling	108
6.2	User Interaction Levels	109
7.1	Survey Results of Chronicles of Singapura in Catholic High School	132
8.1	Different User Preferences Cause Different Requirements	141

CHAPTER 1

INTRODUCTION

1.1 Interactive Storytelling

A story is defined as “an account of an event or series of events, real or imaginary” [64]. Storytelling is defined as “the act or practice of telling stories” [66], it is the art of conveying events of story in words, images and sounds to the audience [65]. It is widely used in everyday human communication, allowing people to share information and convey knowledge of history or experience. The sequence of events is called the storyline (also known as the *plot*) of a story, and it involves the acting of various actors in each event. A typical story includes context, conflict and resolution. Through the act of resolving the conflict and achieving the resolution by the protagonist, values are conveyed to the audience, i.e., “the moral of the story”. Conventional stories have a lot of forms, e.g., movie, drama, cartoon, TV play, radio and so on. They are presented non-interactively, i.e., the audiences are passive receivers who can only watch and listen, but are not capable to determine or affect the routine of the storyline dynamically.

With the emergence of new media and new technology, interactive storytelling has now been made possible. The term *interactive storytelling* is defined as, a form of interactive entertainment in which a player adopts the role of protagonist in a dramatically rich environment [17]. Here interactivity refers to the ability of the audience to be involved in

the story world and interact with the characters or other entities. It empowers the audience to create a personalized storyline, and thus enhances the interest in the storytelling greatly. Interactive storytelling enables the audience to interact with the director or the actors and change the storytelling progress. With the help of virtual reality technology, digital storytelling (or virtual storytelling) has advanced rapidly in recent years. Designers are able to create very realistic or very fascinating world with interactions, bypassing the limitations of traditional storytelling platforms (e.g. theaters). Interactive storytelling in the virtual environment (i.e. interactive virtual storytelling) allows the users to participate the storytelling actively in the virtual environment.

Currently, there are several main research directions of interactive storytelling: one is plot-based interactive storytelling which focuses on interactive story generation, e.g., Mimesis [76], Universe [41], and Brutus [6]. Another approach is character-based interactive storytelling which focuses on interactive story presentation, e.g., Interactive Storytelling [14], TALE-SPIN [50], and Oz project [46]. Moreover, a hybrid approach of the two tries to find the balance between the two approaches, e.g., Façade [48], MINSTREL [68], and Virtual Storyteller by Thénue [34].

1.2 Motivation

Interactive technologies turn observers into participants in many application domains, from home entertainment to cultural exhibits to educational methodologies to personal computing [29]. Interactive storytelling empowers storytelling with user participation, which enhances the audience’s experience eventually.

Interactive storytelling in the virtual environment has attracted a large amount of research interest. There is a proverb, namely, “there are a thousand Hamlets in a thousand people’s eyes”. Different audiences have different perspectives and interpretations of the same story. Each audience has his/her own preferences and opinions of the story and characters, thus he/she prefers to experience a personalized storytelling rather than a fixed one. Conventional media (e.g. theater) cannot achieve such personalized storyline; however, virtual

reality provides an environment in which a story with many variations can be easily and successfully delivered. 3D movie “Avatar” by James Cameron receives amazing welcome over the world for its immersive experience. However, is it possible to act as Avatar “Jake”? As a mix of storytelling, user interactions and virtual reality, interactive virtual storytelling provides the audience the ability to interact with the storyteller or the characters in real-time, such that dynamic and nonlinear storylines which are personalized according to the audience’s intentions/choices can be achieved. An agent is an autonomous, goal-oriented object. Due to its characteristics of autonomy and reactivity, it has already been widely used to simulate the virtual characters, storyteller. With the help of agents, the story generation and story presentation can be done automatically and intelligently with respect to the audience interactions and the context changes.

In my definition, *interactive storytelling* is a form of interactive entertainment in which a player participates the storytelling in a dramatically rich environment by acting as or interacting with the storyteller, a protagonist or an observer.

1.3 Problem Statement

Interactive virtual storytelling attracts many research interests. However, there are not many interactive storytelling systems delivered successfully to facilitate interactive virtual storytelling development. Compared to conventional storytelling, there are a number of challenges, which include:

- The balance between story plot and characters is very important [57]. Story plot and character are the two most important elements of a successful story. The story plot needs to be interactive and includes meaningful alternatives for the audience to play with, in order to achieve different results (i.e. conveying different values). Moreover, as the storytelling is presented in a virtual environment, the performance of the virtual actors affects the audience’s experience directly. Therefore, there is a need to ensure the robustness of the plot at the high level as well as the convincing performance of characters at the low level.

- Numerous model tools have been implemented to model the behaviors/tasks for the director or characters in the storytelling, e.g., Hierarchical Task Network (HTN) and Schema. However, behavior autonomy only represents low-level intelligence of agents. There is a lack of modeling tool to handle high-level goal autonomy, which is a key problem at improving the agent believability.
- Currently, there is a lack of full-fledged interactive storytelling engine and methodology for non-technical designers (e.g. teachers) to develop environments and interactive stories quickly and reliably. There are some architectures implemented with typical scenarios. However, they cannot be easily used to create new scenarios.
- “Choice” is the most important factor in providing a personalized storytelling to the audience. The storyteller needs to make reasonable “choices” to select a proper storyline, and characters need to make “choices” on how to act with respect to user interactions and environment changes. Making a reasonable “choice” is a challenging problem, as the context changes and user interactions can be dynamic and complex, and the context information may be incomplete. Therefore, intelligent storytellers with cognition ability are required to handle all the user interactions and context changes.

1.4 Objectives

The objective of this research is to build a full-fledged agent augmented interactive methodology to design interactive storytelling in a virtual environment. Based on goal-oriented agents, the storytelling system should be able to present the stories and provide an interactive experience intelligently in the virtual environment. It should be robust and efficient with easy and fast development even for non-programmers. Context awareness and user awareness are two main features of the agents with help of strong reasoning capabilities. High-level story dynamics and low-level character performing dynamics are achieved, which enables the audience to experience different levels of interactions with director or characters

by modeling the goals rather than tasks of agents. Moreover, the personality and emotion are also modeled to make the agents more believable, thus presenting a more engaging story.

1.5 Summary of Contributions

In this thesis, we address a new agent-oriented methodology for interactive virtual storytelling. In this approach, interactive storytelling becomes the process of goal pursuing of a director agent and various character agents, which are modeled with an efficient Fuzzy Cognitive Goal Net model. Thus, users are able to interact with both the director agent and character agents in real-time. The main contributions are summarized as follows.

(1) **Agent-oriented Methodology for Interactive Storytelling**

A new interactive storytelling methodology based on goal-oriented agents (AOMIS) has been proposed, that bridges the gap between plot-based and character-based interactive storytelling with goal-oriented agents. In this approach, both the story plot generation and character performing generation are realized in real-time as goal pursuing of agents. Moreover, both story coherence to convey a value and believability of character performance can be achieved. By acting as or interacting with one of the agents (e.g. director and character), the audience is able to experience the storytelling with multiple degrees of freedom. Compared to action-oriented agents in other researches, goal-oriented agents exhibit higher believability with both goal automation and action automation.

(2) **Fuzzy Cognitive Goal Net**

Fuzzy Cognitive Goal Net (FCGN) model has been proposed as the story plot model and the character model. In proposed interactive storytelling methodology, FCGN is used to plan plots for the director and to schedule behaviors for the characters. Fuzzy Cognitive Maps (FCMs) are used as the context model to model the dynamic causal relations among context variables, user interactions, and agent decisions.

The promising fuzzy cognitive ability empowers the agents to reason in a complex, incomplete, and dynamically changing context.

(3) Evolutionary Fuzzy Cognitive Map

A believable virtual world has dynamic and evolving causal relationships among the elements in the world context, while Fuzzy Cognitive Map is not capable to model it. Evolutionary Fuzzy Cognitive Map has been proposed as the context model to model the probabilistic and fuzzy causal relationships among the context concepts, thus providing the basis of agent reasoning in order to provide personalized storytelling.

(4) An interactive storytelling engine (S-MADE) to create interactive storytelling from scratch

An interactive storytelling engine, namely Multi-Agent Development Environment for Interactive Storytelling (S-MADE), has been constructed to implement interactive storytelling agents with predefined story scenarios. It enables designers without programming skills to develop interactive storytelling scenario conveniently.

(5) Apply agent-based interactive virtual storytelling methodology to promote teaching and learning

Agent-oriented interactive virtual storytelling are created with our methodology for the science education to promote teaching and learning. By embedding the learning contents into the story, the students were able to obtain a deep learning experience through their explorations and interactions with story characters. Agent-based interactive storytelling promotes the learners' interest as a good reinforcement to classroom learning.

1.6 Thesis Structure

The following of the thesis is structured as follows:

- ***Chapter 2 Theoretical Background and Related Work***

In this chapter, the related work of interactive virtual storytelling as well as the theoretical background blue are presented. In related works section, recent research and developments in interactive storytelling are reviewed; in addition, the features and limitations of current approaches are investigated. In the theoretical background section, agent technology and the soft computing models used by agents (i.e. Fuzzy Cognitive Map and Goal Net) are discussed.

- ***Chapter 3 Agent-oriented Methodology for Interactive Storytelling (AOMIS)***

In this chapter, we propose a goal-oriented agent-based hybrid interactive storytelling methodology. In the methodology, the interactive storytelling is modeled as a multi-agent system with goal oriented agents, which includes the director agent and character agents. The interactive storytelling is generated as goal pursuing of the agents by interacting with the players. By interacting with the director or character, the player can either change the storyline or change the behaviors of the characters.

- ***Chapter 4 Fuzzy Cognitive Goal Net (FCGN)***

The intelligent agents are goal-oriented; therefore, an efficient goal model is required to model the goals of the agents. In this chapter, we propose the Fuzzy Cognitive Goal Net (FCGN), which models the agent goals as a hierarchical goal network, with fuzzy reasoning about user interactions and story contexts. Moreover, FCGN modeling to model the story plot and the actor behaviors is illustrated.

- ***Chapter 5 Evolutionary Fuzzy Cognitive Map (E-FCM)***

In order to provide a customized storytelling to players in real-time, we propose Evolutionary Fuzzy Cognitive Map to model the complex causal relationships among story scenes, user preferences, and real-time interactions.

- ***Chapter 6 Interactive Storytelling Engine to Create Interactive Storytelling from Scratch***

In this chapter, an interactive storytelling engine S-MADE is illustrated to create an interactive storytelling from scratch based on the agent methodology. Non-technical users (e.g. teachers) can generate a workable interactive storytelling scenario conveniently with simple drag-and-drops. The interactive storytelling will then be conducted automatically.

- ***Chapter 7 Applications of Interactive Virtual Storytelling for Education***

Interactive storytelling has been successfully used in the education, and proven to promote learning interest significantly. In this chapter, three projects about serious games and interactive storytelling are introduced.

- ***Chapter 8 Conclusions and Future Recommendations***

In the last chapter, conclusions are made on the agent-based hybrid interactive storytelling approach and a full methodology to implement the interactive storytelling. Moreover, future refinements on our current methodology and tools are also recommended in order to improve the work flow and usability further.

CHAPTER 2

LITERATURE REVIEW AND THEORETICAL BACKGROUND

2.1 Introduction

Throughout the human history, storytelling is an important practice in conveying knowledge. Interactive storytelling is now possible with the aid of new media (e.g. virtual worlds), allowing audiences to actively contribute to the storytelling. In this chapter, we review the state of the art of storytelling, interactive storytelling in conventional environments as well as that in the virtual environments. Moreover, we introduce the theoretical background of interactive storytelling, with regard to agent technology, goal net model as well as a soft computing model Fuzzy Cognitive Map, which are the basis for our agent-oriented interactive storytelling methodology.

2.2 Story, Storytelling, Interactive Storytelling and Virtual Storytelling

2.2.1 Story and Storytelling

Stories are used to transmit information throughout human histories.

$$Context + Conflict \Rightarrow Resolution \quad (2.1)$$

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

The formula shows the essence of a simple story. The protagonist in certain context meets certain conflict. After he/she overcomes the conflict, there is a resolution at last. The typical story elements are:

Protagonist

The protagonist is the main character who is involved in the story to convey the story value. A story may have one or more protagonists.

Context

The story events are only meaningful within certain contexts. Context refers to the background information of the story, which is the foreshadowing for the conflicts of story. It includes the time, state and event for the story. For interactive virtual storytelling, the context is more complex as the player is immersed in the virtual environment. The presence of the player with regard to other entities is an important part of the context.

Conflict

Conflict is a must that makes story from narrative. The protagonist needs to resolve the conflict in order to reach the resolution. The conflict needs to be designed properly, so that the protagonist could resolve it with a “change” of mind or ability in a convincing manner.

Resolution

The resolution is the final achievement of peace after the protagonist solves the conflicts.

Storytelling is the process of presenting the stories, that is the process of showing how the protagonist resolves the conflicts and reaches the resolutions. Stories are not only transmitted in the form of narrative, but also in other forms like pictures, dramas, movies, and cartoons etc. The emergence of the new media has stimulated development on storytelling greatly. Digital storytelling is a new form of storytelling in which stories are presented in

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

a digital context (e.g. web, computer game, and virtual world). Virtual storytelling is a type of digital storytelling which handles the storytelling in the 3D virtual environment. It has become increasingly popular as the emergence of several kinds of technology (i.e. storytelling, multimedia, and computer) makes a personalized and immersive storytelling. In *Poetics* [1], Aristotle (330BC) concludes six components of tragedy, which are shown in Figure 2.1 with decreasing importance. *Plot* is the most important element of a successful story. It is defined as “the arrangement of the incidents”. Figure 2.2 shows the relationship of story tension with respect to time in different stages of story plot, from exposition with the introduction of contexts, to ascension where the conflicts are presented, then to climax where the conflicts are resolved, before going lastly to conclusion. *Characters* are the main objects of representation. They perform their moral roles through speeches as well as actions assigned by the director. *Thought* represents the values or morals to be conveyed from the story which could be educative or entertaining. *Diction* is the way language is used to convey the representation. *Song* is an embellishment of language and *spectacle* is an embellishment of performance.

Figure 2.1: Six Components of Tragedy (Aristotle, 330BC)

2.2.2 Interactive Storytelling and Virtual Storytelling

Conventional storytelling methods are unidirectional, i.e., the audiences are forced to either accept the story, or give up the story. However, there is a demand of the audiences to join

Figure 2.2: Tension vs. Time at Different Story Stages

in the storytelling process actively to create a personalized storytelling. For example, they may wish to feedback their feelings about the characters and events to the storyteller, and then the storyteller is able to adjust the storyline accordingly. Interactivity enables both the storyteller and audience to affect the story progress. Also, the audiences are able to query information of the story through interactions, which promotes understanding and clarifies possible questions that the audience might have.

Interaction brings immersion, participation, a sense of control, and freedom. Thus, increasing the interaction meaningfully leads to an increase in believability, feedback, and degree of freedom. Hoffmann et al. [35] expanded the conventional storyline to a dynamic storyline in interactive storytelling, and shows that the new storyline broadens the experience in the dimensions of time and content.

Depending on the modeling method, interactive storytelling can be divided to plot-based, character-based, and hybrid interactive storytelling.

2.2.3 Plot-based Interactive Storytelling

Plot-based interactive storytelling, which is author-centric, focuses more on interactive story authoring or generation. It concerns the structure of the story, i.e., coherence of the story scenes, rather than the refinements of characters behaviors.

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

Lebowits [40, 41] from Columbia University developed the *Universe* system, to generate the stories without fixed endings, such as in soap dramas. The *Universe* system defines a series of scenes, which can be achieved with certain pre-conditions. A controller connects the scenes with fulfilled preconditions together to form the story.

Young et. al. [75, 76] from Liquid Narrative Group of North Carolina State University developed a component-based architecture called *Mimesis*. It integrates plan-based behavior generation with an interactive game environment as shown in Figure 2.3. It can communicate with different platforms, PDA, handphone, and web browser etc. The *Mimesis* system can generate structural interactive behaviors in the virtual worlds. It defines a multi-agent system and designs a virtual environment control mechanism. Intelligent agents are used to realize high level narrative and user interactions.

Figure 2.3: Storytelling Architecture of Mimesis Project

Brutus [6] was developed by Ferrucci as an architecture for automatic story generation. The system is built on a Prolog-based system that allows the defining of frame-structures, relations between frames, and production rules, all in an English-like format.

Pérez and Sharples [74] developed MEXICA to generate stories with two phases of Engagement and Reflection. The system adopts elements from existing stories (engagement) and generates a new story (reflection).

Magerko et al. [44] proposed an Interactive Drama Architecture (IDA) which employs a story director agent to manage the user's narrative activities.

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

Making the storytelling agent more believable is very important. Natural language processing and strong inferential capabilities are two significant features. The VISTA project by Elizabeth F. et al. [28] creates an agent architecture to conduct virtual interactive storytelling. In the architecture, pre-recorded stories are organized in hierarchies, and the story information is indexed in the AIML scripts. In front end, the storytelling agent interacts with users through natural language queries/answers. In the back end, a knowledge server provides inferential capabilities to the story contents.

While interactive storytelling adds to the enjoyment of the user, it also increases the complexity of the system, as the user interventions cause the story scenario to change. Therefore, there are many schemas used in the current interactive storytelling in order to prevent or restrict problems which may arise due to these changes. For instance, the user could be a spectator in the virtual storytelling world, who is transparent to other roles in the games. Although they are allowed to move some objects in the virtual world to make the storytelling more dynamic, the whole story scenario will not be affected. Users can experience the virtual worlds through the eyes of virtual identities, i.e., he/she can experience different views rather than an identical one [32].

Besides the research on the architecture of story generation or presentation, some products were created for the development of interactive storytelling scenarios. Crawford et. al. [17] developed Storyton for creating interactive storytelling. Crawford considers that “verb thinking” is very important to a successful interactive storytelling design. With Storyton, the designer is able to design an interactive story scenario easily with predefined “verbs”. Similarly, Alice from the Carnegie Mellon University is developed for non-programmers to build 3D interactive storytelling environment easily and quickly. Designers are enabled to define events for the interaction, e.g., mouse clicking or key pressing, and also track the variable change.

As a conventional approach, plot-based interactive storytelling shows interactions between the audience and the director, who ensures the coherence of the story plot, so that the story scenes are consistent in conveying the story merit. Interactive storytelling in a virtual environment requires the behaviors of the virtual characters to be believable and personalized,

whereas the plot-based approach cannot ensure this as it concerns only high-level story generations.

2.2.4 Character-based Interactive Storytelling

If we consider plot-based interactive storytelling as a centralized approach, then character-based interactive storytelling is a distributed approach. By setting the behaviors of the characters with constraint conditions, the characters interact with each other to form the story. In this approach, the character believability is one main concern.

TALE-SPIN [50] is a very early story generation system, made by Meehan from Yale University. The users will first assign roles, goals, and behaviors to the characters. Every character will perform its behaviors and interact with each other until it reaches its goals. Based on this, they make changes according to other users changes and context changes in the world with reasoning and planning. Thus the story is formulated.

As an early practice, the Oz project lead by Mateas from Carnegie Mellon University has an influence over study of interactive drama and believable agents [46, 47]. This system allows users to author an interactive drama, which includes a virtual environment, a group of characters, a virtual representative of an user, a narrator, and a drama manager. The user interacts with the characters through the virtual representative. At the same time, the drama manager coordinates whole interactions. Characters are modeled with believable agents. They are reactive, goal-oriented, emotional, intelligent, and can use natural language to communicate with the user.

Marc Cavazza's group from Teesside, UK [14, 15] made a great contribution towards character-based interactive storytelling, in which each character has its own role and actions, and the characters have dynamic interactions between them based on guidelines from the storyline. In order to describe the complex goal of a character, Hierarchical Task Network (HTN) is used to represent the goals or tasks for each character. To improve the quality of real-time planning of HTN, anytime planning was proposed by Nick Hawes [33] to replan in a complex environment in real-time. Compared with plot-based interactive sto-

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

Figure 2.4: Hierarchical Task Network for Character Action Planning

rytelling, character-based approach generates the storytelling from free interactions among the characters rather than from a fixed story plot. Though it exhibits the believability of the characters, allowing low-level interactions with the users, the storytelling might not preserve good coherence of story, which might result in the merit of the story being distorted.

2.2.5 Hybrid Interactive Storytelling

Besides the previous two approaches, there is a hybrid branch that does not focus exclusively on the plot or character, but a mixture of the two. Both the structural plot and believable characters are emphasized simultaneously.

Façade is currently one of the most successful interactive storytelling systems by Metas [47] [48]. It overcomes the plot incoherence problem from Oz project. A screenshot of Façade is shown in Figure 2.5. It integrates plot, character, and natural language processing together to achieve a successful interactive drama. In their system, a simple drama manager is implemented to plan the storyline based on a reaction formula of the scenes.

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

The story builder assigns to each scene an individual tension value that is applied to select the succeeding scene. Each succeeding scene before the climax should have more tension than the previous one. After the climax, the drama manager switches to select a scene with less tension. Moreover, the characters in the virtual environment show very human like behaviors and emotions, which help greatly with engaging experience to the users. Tuner from University of California developed MINSTREL [68] to generate the story of

Figure 2.5: Screenshot of Two Characters in Façade [48]

“King Arthur”, it makes the computer model of agent schema for story generation. In MINSTREL, all the elements of story are defined with Schema, including author schema for story plot and character-schema for characters. Author schema defines the author level plan, which guides how the story will flow from beginning to conclusion with good coherence. Character schema is the character-level goal to fulfill the characters’ desire, intention, and status. MINSTREL predefines six kinds of planning advice themes (PATs) to generate the structural plot of the story. The story generation is achieved through realization of author-schema and character-schema. MINSTREL also defines TRAMS (transform recall adapt methods) to transit the scenes in order to generate a new and fitting scene.

Virtual Storyteller by Thenue [34, 67] is constructed as a multi-agent system. The system simulates a virtual music center, includes virtual director, virtual characters, and a narrator. Virtual director generates the story plot. Characters and the director can reason on

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

their own knowledge bases, while a narrator agent generates the story text with natural language, and a speaking presentation agent narrates the story text with gestures and language.

Cao et. al. [12] proposes an intelligent narrative and animation architecture, namely PNAI. The architecture is constructed as a multi-agent system, including the director agent and characters agents. Narlog (Narrative Logic) is created to control the agents' knowledge and behavior.

Compared to plot-based and character-based approaches, the hybrid approach combines the advantages of the two approaches, such that both the compelling story plot and the believable characters are realized with this approach.

However, current hybrid approach still has some problems. For example, director and characters are modeled with low-level behavior autonomy, which show only limited intelligence at handling the user interactions and context changes. Moreover, how the director assigns the scene to the characters in real time is a key challenge for story generation and presentation automation. The main reasons for the two problems are a lack of storytelling architecture allowing high-level goal modeling and a lack of goal model that can be used by the director and characters to communicate.

2.2.5.1 Interactive Storytelling Authoring Systems

There is a need to develop interactive storytelling authoring systems to facilitate the interactive storytelling authoring, especially for the novice users. Alice provides a graphical user interface to create a virtual environment with virtual characters, it is suitable to learning basics of programming. Storytelling Alice is created based on Alice, to allow to create storytelling by Kelleher [37].

Moreover, Storytron is launched to create interactive stories by Chris [17].

However, to the best of our knowledge, there is a lack of authoring tools to develop interactive storytelling and deploy into different 3D virtual environments, as different users might be familiar with different virtual environments.

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

Figure 2.6: Graphical User Interface of Storytelling Alice

2.2.6 Narratology

Narratology denotes both the theory and the study of narrative and narrative structure and the ways that these affect our perception [3]. Narratology research can be traced to Aristotle (Poetics), while modern narratology starts from the Russian Formalists, particularly Propp [55]. Propp analyzed the basic plot components of Russian folk tales to identify the simplest inducible 31 narrative elements, that provide the basis for the interactive narrative research.

2.3 Theoretical Background

2.3.1 Agent

In the current research of interactive storytelling and narratology, agent technology has become a promising solution to model the intelligent director, characters, and other roles used in the interactive storytelling. It brings the intelligence, automation, and robustness, which are important for the story development and the interactive user experience in the interactive storytelling. However, agents have not been explored systematically in the current interactive storytelling systems. Therefore, this section will give a brief review of agent properties and architectures, which are the foundations of our research in the domain

of interactive storytelling.

2.3.1.1 Agent Properties

The term *agent* has been used frequently in many research fields, for example, *learning agent*, *intelligent agent*, *mobile agent* etc. However, there are different views on the definitions of *agent*. An *agent* has been defined as “persistent software entity dedicated to a specific purpose” [62], “computer programs that simulate a human relationship by doing something that another person could do for you” [58], “active object” [61], and “smart object” [54] etc.

Due to the diversity of agent usage, it is very difficult to give a unified definition of agent. Instead, it is more reasonable to characterize and categorize agents along certain dimensions. Wooldridge and Jennings [73] provide one of the most famous and foundational classification of agent’s characteristics. They distinguish agent’s characteristics into a *weak notion* and a *strong notion* according to the usage of agents. The weak notion of agency is identified by four properties: *autonomy*, *reactivity*, *pro-activeness*, and *social ability*.

Autonomy: An agent can control its actions and internal states, and it can operate without the interventions of a human being.

Reactivity: An agent can perceive its living environment, and it can respond to the environment changes.

Pro-activeness: Besides responding to the environment changes, an agent can exhibit some goal-oriented behaviors by taking initiatives.

Social Ability: Agents can interact with each other via some kinds of agent communication languages.

Emotional: Agents have emotions to show their internal states.

The weak notion identifies the most fundamental characteristics of an agent. The strong notion is the higher level of weak notion. In the strong notion, an agent should have

the properties listed in the weak notion. Furthermore, it should also have other high-level capabilities that are possessed by human being, such as knowledge, belief, intention, desire, and emotion. BDI (believe, desire and intention) agent [56] is one example, which describes an agent as an entity with belief (knowledge), desire (goals of agent), and intension (actions taken to meet the desire).

2.3.1.2 Agent Architectures

Agent architecture defines how to build or construct a software entity that has agent properties. It specifies how an agent can be implemented with a set of interactive components. In the same way, it provides some guidelines on interactive storytelling architecture design. In general, agent architecture can be classified into three categories:

- **Deliberative Architecture**

Deliberative agent architecture is a classical agent architecture. Jennings defines a deliberative agent as one “that contains an explicitly represented, symbolic model of the world, in which decisions are made via logical or pseudo-logical reasoning, based on pattern matching or symbolic manipulation” [73]. This kind of agent has a mental state that reflects its living world, and it is able to make reasonable decisions to handle its actions. Compared to the reactive agent, which will be described later, a deliberative agent is more active and autonomous as its decision making is based on internal reasoning rather than an event triggered by the outside environment. A number of deliberative agent development attempts have been made and some deliberative architectures have been built, such as Intelligent Resource-Bounded Machine Architecture (IRMA) [5], HOMER [70], GRATE [36] etc. Several of them have implemented the well-recognized Belief-Desire-Intention (BDI) agent model [56].

- **Reactive Architecture**

Reactive architecture is an alternative of deliberative architecture. As its name indicates, this architecture emphasizes the interaction between an agent and its environment. Instead of focusing on complex reasoning and symbolic manipulation, a

reactive agent acts using a stimulus/response type of behavior, responding to the present state of its living environment [26]. This does not mean that a reactive agent is not an intelligent agent. Brooks has argued that the “real” intelligence is situated in the world and intelligent behavior arises as the result of an agent’s interaction with its environment [7, 8]. Agre also pointed out that most everyday activity is “routine” in the sense that it requires little (if any) abstract reasoning [16]. Many systems have been successfully developed using various reactive architectures such as Brooks’s Subsumption Architecture [7], Agre’s routine architecture [16], Mase’s Agent Network Architecture (ANA) [43].

- **Hybrid Architecture**

Many researchers have suggested that neither a completely deliberative architecture nor a completely reactive architecture is suitable for building agents [73] in some complex systems. Hybrid agent architecture is one that integrates deliberative architecture and reactive architecture with their advantages. In general, there are two essential components inside a hybrid agent: a planning or reasoning component to control the agent’s long term goals and decisions, and a reaction component to quickly react to the environment changes. Essentially, these two components can be considered as the representations of deliberative architecture and reactive architecture in the hybrid agent respectively. Hybrid architecture has been widely used and a number of such architectures have been designed, for example, Procedural Reasoning System (PRS) [30], TouringMachines [27], and COSY [9].

2.3.2 Goal Net

An intelligent agent is not only reactive, but also proactive in making decisions. It is goal-driven, and plans its behaviors based on its goals. Goal net is a goal model proposed by Shen [78] to model complex goal through hierarchical planning and decomposing. It has been successfully used in multi-agent system modeling in business forecasting and E-learning [59]. Goal net model is composed of two main components: *goals* and *transitions*.

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

As shown in Figure 2.7, *goals*, denoted as circles, are used to represent the goals that an agent pursues. *Transitions*, represented by arcs and vertical bars, connecting from the input goal to the output goal, specify the relationship between the two goals. Each transition is associated with a task list which defines the possible tasks that the agent needs to perform in order to transit from the input goal to the output goal.

There are two kinds of goal representations in Goal Net, *atomic goal* and *composite goal*. An *atomic goal* is a primitive state which cannot be further divided, while a *composite goal* is an abstract goal that can be split into sub goals connected via transitions. Therefore, a complex goal can be recursively decomposed into sub-goals and sub goal nets until only atomic goals are left. The hierarchical structure simplifies the goal modeling process with different levels of abstraction. In Goal Net, there are four types of temporal relations of

Figure 2.7: A Sample Hierarchical Goal Net

goals represented by transitions: *sequence*, *choice*, *concurrency*, and *synchronization*, shown in Figure 2.8. The transitions show different causal relationships of goals as below:

Sequence: A direct sequential causal relationship between input goal i and output goal j

Choice: A selective connection from input goal i to possible output goals j and k , and only one output goal can be selected based on selection criteria

Figure 2.8: Goal Net Transitions for Goals G_i , G_j and G_k : (a) Sequence (b) Concurrency (c) Choice (d) Synchronization

Concurrency: Upon achieving input goal i , all the output goals j and k can be activated simultaneously

Synchronization: A synchronization point from different input goals i and j to a single output goal k , and the output goal can be achieved only when all its input goals are achieved

In a Goal Net model, a goal is represented as S_i , and the transition is represented as T_i .

2.3.3 Fuzzy Cognitive Map

A Fuzzy Cognitive Map (FCM) is a kind of qualitative modeling tool proposed by Kosko [38]. It provides a simple and straightforward way to model causal relationships among different variables. Currently, FCMs have been widely used in many real-time applications. For example, FCMs was used to model the movements of the sheepdog and sheep [45]. Kosko and Dickerson [21] used FCMs to model the hunting process among sharks, dolphins, and fishes. Besides the fuzzy cognition of the FCMs, an extension, namely Probabilistic FCM, was proposed by Song [63] to allow simulating probabilistic events.

FCM includes two elements: *concepts* and *causal relationships*. An example is shown in Figure 2.9. *Concepts* are represented as circles, which denote the related causes and effects

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

in the model. The concepts are represented as fuzzy values. The *causal relationships* are represented as directed arcs, each of which has a sign and a weight. The sign shows the causal relationship between two concepts, which is compulsory. The '+' sign means 'positive causal relation', in which the increase of the starting concept value will result in the increase of the ending concept value. The '-' sign means 'negative causal relation', in which the increase of the starting concept value will cause the decrease of the ending concept value. For example, increase of 'bad weather' causes increase of 'freeway congestion'; on the other hand, 'freeway congestion' causes decrease of 'own driving speed'. The weights differentiate the significance of causality from the start concept to end concept, and they are represented as fuzzy terms as well. For example, 'bad weather' usually causes 'auto accidents'. The term 'usually' is a fuzzy item used to describe the level of significance. If there is no arc linking two concepts, it means that there is no causal relationships between the two concepts, i.e., the two concepts are independent. A Fuzzy Cognitive Map represents

Figure 2.9: A Sample Fuzzy Cognitive Map (Bart Kosko)

a knowledge base in a domain about how concepts relate to each other. Knowledge from different experts can be accumulated through combining several FCMs into a big FCM. This is done by merging the same concepts. A Fuzzy Cognitive Map provides qualitative

CHAPTER 2. LITERATURE REVIEW AND THEORETICAL BACKGROUND

information about the relationships. It ties facts, things, processes to values and policies and objectives, which makes it possible to predict how complex events interact and play out.

To facilitate the analysis of a FCM, each concept is represented as a state value whose range is $[0, 1]$ or $[-1, 1]$, while the causal relation is represented as a weight, whose range is $[-1, 1]$. Suppose the current value of concept i is A_i , its new value can be updated as

$$A_i^{t+1} = f(k_1 \sum_{j=0}^n A_j^t \cdot w_{i,j} + k_2 \cdot A_i^t) \quad (2.2)$$

where $f(\cdot)$ is the activation function and A_j is the state value vector. As there might be loops involved in the FCM, the values of concepts will evolve to an equivalence point or a limited cycle in the long run.

2.4 Summary

In this chapter, we have shown an overview of the state of art about the interactive storytelling methodologies and models. Agent technology has been widely adopted in different interactive storytelling approaches due to its promising properties in handling interactions. According to the storytelling modeling method, the interactive storytelling research can be classified as plot based, character based or hybrid. However, to the best of our knowledge, it is rare to find approaches which model the storytelling in terms of intentions or goals of the agents, while most of them model the storytelling in terms of tasks or behaviors. A goal-oriented agent is more intelligent than a task-oriented agent at providing human-like interactions and immersion.

In the next chapter, an agent-oriented methodology for hybrid interactive storytelling is proposed. The storytelling system is constructed as a multi-agent system, which is composed of scriptwriter agent, director agent, and character agents that are all goal-oriented. The agents are modeled to provide different kinds of interactions with the user to create a personalized storytelling.

CHAPTER 3

AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

3.1 Introduction

In interactive virtual storytelling, characters perform the story visually, and so the performance of the characters determines the audience's experience directly. Plot-based interactive storytelling ensures the coherence of story plot, but it is weak at realizing fleshed out believable virtual characters Universe, e.g., [40] Mimisis [76], Brutus [6], MEXICA [74], IDA [44], VISTA [28]. In contrast, current character-based interactive storytelling focuses on the modeling of believable characters, e.g., TALE-SPIN [50], Oz [46], Character-based Interactive Storytelling [14]. However, this approach is only suitable for presenting simple stories. The story generated from characters might be very tedious and would not convey a story merit efficiently. Therefore, a hybrid approach is required to bridge the gap between the plot-based approach and character-based approach for interactive storytelling.

A successful interactive storytelling should preserve the following properties:

- (1) **Autonomous:** The storytelling can be generated and presented autonomously.
- (2) **Believable:** The director selects rational storylines, and the characters can perform believably.

- (3) **Intelligent:** The agents should be conscious of the audience and context changes and be able to make rational decisions.
- (4) **Robust:** The storytelling should be robust to audience interactions and dynamic context changes.
- (5) **Efficiency:** The storytelling should be presented efficiently, in real-time.

Goal-oriented agents present human-like thoughts and behaviors, which are suitable to demonstrate intelligent director and virtual character agents. Though agents have been widely used in previous researches, only some basic properties were explored (e.g. reactive, behavior autonomy), which was not sufficient to present an engaging experience to audiences. One successful hybrid interactive storytelling is Façade [48]. However, there is a lack of a formal methodology with models and toolsets for novice to develop hybrid interactive storytelling easily.

Therefore, **Agent-Oriented Methodology for Interactive Storytelling (AOMIS)** is proposed to create interactive storytelling in the virtual environment from the text-based story. In this methodology, a novel goal-oriented hybrid model for interactive storytelling has been designed, enabling both story plot coherence and character believability to be achieved in real time. The interactive storytelling system is constructed as a multi-agent system (MAS). The interactive storytelling process is carried out through goal planning and execution (more than task planning) by the intelligent agents in the system, which will be illustrated in details in this chapter.

With the proposed model, story generation and presentation are not separated but interwoven. Both the director and characters, interacting with audience, determine the ongoing of the storytelling.

3.2 Agents and Goals in Interactive Storytelling

The storytelling process is achieved through goal pursuing of the agents that are involved in the system. As shown in Table 3.1, there are three typical agents involved in the interactive

Agents	Goals
Scriptwriter	Prepare the storylines
Director	Direct the storytelling
Character	Act the storytelling

Table 3.1: Agents and Their Goals in Interactive Storytelling

storytelling, which are scriptwriter agent, director agent and character agents. The goal of a scriptwriter agent is to create a story plot with multiple meaningful storylines which agree with user expectations. The goal of the director agent is to direct an interesting story to the audience according to user interactions (i.e. to show the story elements), e.g., contexts, conflicts, and resolutions. Moreover, the goals of the character agents are 1) receiving the tasks assigned by the director agent and 2) performing the tasks accordingly.

An efficient and unified goal model is required to model the complex goals and goal relations for the agents mentioned above, in order to achieve the autonomy of story generation and story performing. In our hybrid agent-based interactive storytelling system, Fuzzy Cognitive Goal Net (FCGN) is proposed [10]. It adds the fuzzy cognition capability over the generic goal net, and so is more suitable to model the goals of agents with real-time goal updates and complex user interactions.

The scriptwriter generates the story plot in the form of goal net and the director selects story scenes as one storyline from the goal net with fuzzy cognitive goal selection. Through the behavior dispatching mechanism, the director assigns actors the behaviors which are involved in story scenes as goals dynamically, in the form of goal net.

3.3 Storytelling Elements Definitions

In order to model the interactive storytelling process, the key elements of storytelling need to be defined semantically at the first step.

Definition 3.1 *A scene is an atom of a story plot, which involves the interactions among a number of virtual actors with dedicated roles, in certain context. It can be expressed as a*

CHAPTER 3. AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

tuple as

$$S = [AC_1, AC_2, \dots, AC_n, T_1, T_2, \dots, T_n, C]$$

where AC_1 to AC_n are the actors involved in the scene, T_1 to T_n are the behaviors which are assigned to each actor respectively, C is the context of the scene.

Definition 3.2 *An actor is the virtual entity which performs the behaviors assigned by the director. It is a hybrid of goal-driven and reactive agent, which means that it both has goals to pursue and it is reactive to events. Each actor has its own profile, which includes states, preferences and actions that can be performed. It is expressed as*

$$AC = [S_t, P, B]$$

where S_t is a list of states including name, position etc. P is a list of preferences, and B is the behaviors of the actor which is empty initially and assigned by the director agent in the storytelling process in real-time.

Definition 3.3 *A behavior of a actor is an acting slice assigned by the director in a scene. A composite behavior can be decomposed to a sequence of sub behaviors recursively. A behavior is expressed as*

$$B = [B_1, B_2, \dots, B_n]$$

where the behavior B is composed of a series of behaviors $B_1, B_2, \dots, B_n$ in the temporal order.

Definition 3.4 *Context is the circumstances in which a scene occurs. It is represented as*

$$C = [C_1, C_2, \dots, C_n]$$

where the context C includes a list of states of virtual environment.

Definition 3.5 *Interaction is the activities that an actor can conduct inside the virtual environment. It is represented as*

$$I = [AC, C, B, Tp]$$

where AC is the actor, C is the context, B is the behavior, and Tp is the type of activity.

3.4 Structure Model of Interactive Storytelling

Agents are goal-oriented, autonomous entities which work together in a specific context. In storytelling, director, characters, and other entities (e.g. environment) are autonomous with beliefs and intentions in order to narrate the story. By specifying these elements as goal-oriented agents, the interactive storytelling process can reach a high level of autonomy in terms of story generation and presentation, and save a lot of human effort in tedious implementations when designing scenarios. Interactive storytelling in a virtual environment may involve agents with multiple roles, i.e., a scriptwriter, a director, and virtual characters. The structure model of hybrid interactive storytelling can be viewed as a multi-agent system which is depicted in Figure 3.1.

Figure 3.1: Multi-agent System View for Interactive Storytelling Hybrid System

3.4.1 Agents

The circles in Figure 3.1 denote the agents which are involved in the interactive storytelling system.

Scriptwriter

The goal of a scriptwriter agent is to construct the plot of the story. Different from the traditional story plot, the plot created by the scriptwriter agent is nonlinear

CHAPTER 3. AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

with multiple storylines, providing choices and alternatives to satisfy the different requirements of the audience. Typically, the scriptwriter is the representative of the researcher/designer, who will build multiple meaningful storylines for a typical story scenario.

Director

The director is also known as the dramatist or drama manager. A director agent is the storyteller, who is responsible for selecting and determining the storyline from the whole plot given by the scriptwriter agent. According to the user interactions U as well as context changes E , one reasonable storyline is chosen by the director dynamically. Depending on the complexity of the story presentation, sub-directors may be needed to share the workload.

Virtual Characters/Actors

The virtual actors are the virtual entities required to perform the scenes of the story, which are shown in the box container named as “Actors” in Figure 3.1. Unlike traditional storytelling, the actors with their own specific preferences are vivid entities who perform the acts to audience directly. AC_1 , AC_2 , and AC_3 are three sample actors. They take over the behaviors as acts from the director and perform in temporal sequence. Among the actors, the one who is keen to convey the moral of the story is called *protagonist*. The actors interact with each other according to the plot, and the audience interactions U and context changes E may also affect the behaviors of the actors accordingly.

3.4.2 Audience/User/Participant

In conventional storytelling, audience listens to the story presented by storyteller. In interactive virtual storytelling, the audience takes part in the virtual environment to experience the storytelling with allowable interactions. Therefore, the audience is not just a pure observer, but also a *participant*. The audience can have their virtual representative in

CHAPTER 3. AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

the virtual environment visible or invisible to other actors. According to different roles in interactive storytelling, the audience can freely take on various roles. The term *user* or *participant* is also used interchangeably with the same meaning in this report. The audience are able to join the storytelling as one of the actors (mainly protagonist) or as the director.

User interaction is one of the most significant features in interactive storytelling. In the proposed interactive storytelling system, the audience is able to interact with the virtual characters, the director and the context. As shown in Figure 3.1, $U_1, U_2, \dots, U_n$ are the interactions with the virtual characters, which will affect the decisions and actions of virtual characters. $E'_1, E'_2, \dots, E'_n$ are the interactions with the director, through which the director selects reasonable scenes to be executed by the characters in real-time. Moreover, the audience can send feedback to the scriptwriter for commenting new storylines, thus the scriptwriter agent is able to learn from the feedback and composes new meaningful storylines.

Audience have different preferences, different expectations of story and different personalities. User awareness means that user properties are taken into account in the storytelling, such that a personalized storyline and performance are generated.

In the storytelling, director agent and characters belong to different domains with different properties are shown in Table 3.2. Therefore, the storytelling system can provide higher

Parameters	Director	Character
Role	Coordinating a story presentation	Performing the acts to form the story
Target	Plot	Scene
Knowledge	Global	Local
Action	High Level	Low Level

Table 3.2: Comparison of Director and Character

degree of freedom if the participant is able to act as the director or one of characters.

3.4.3 Story Context

Story context represents a series of circumstances or background variables in which a story scene occurs, and it is story scenario specific. Story scenes are meaningful only under certain context. The context of conventional storytelling is often static and direct, such as by using “once upon a time”. By introducing the status of protagonist and history background, the story reveals to the audience the difficulties of the obstacles and the challenges to reach the resolutions, and indicates the value at last. However in interactive storytelling, the context is dynamic in both values and time. This increases the variability and complexity of the storytelling.

For interactive storytelling in virtual environment, the context might include the following variables:

- **Story background:** every story happens at sometime at somewhere, with or without history
- **Character status:** different characters have different identity, personality, and ability, which determine how they will act in the storytelling
- **Virtual environment:** for digital storytelling, the virtual physical environment is also very important to the virtual actors as it is the drama stage, e.g., weather, land, and availability of virtual objects.
- **System and resources:** besides the story itself, the hardware limitations might also affect the strategy of the storytelling process, level of reasoning, and level of intelligence. Examples include processing power, network bandwidth etc.

In a hybrid system, contexts evolve automatically, and can be invoked by the director and actors with an event. Context awareness is an important property of a robust interactive storytelling system, which means that the director agent is able to generate the storyline and character agents perform the scenes dynamically and believably with the context criteria.

3.4.4 User Interactions

User interactions refer to the activities of the user in the storytelling process, which is the key for the success of interactive storytelling. There is a need to interpret the user interactions and conduct the according changes for the interacted entities, e.g., a director or characters.

The activities could be initiated by the user or by the entity in the storytelling (e.g. director or characters). The activities can be classified into the two types: 1) active activities which alters the goals and behaviors of the director agent or character agents; 2) passive queries that help the user to get more information about the story. The activities include:

- *Acquisition*: the user queries information he/she is interested in from the director agent or character agents.
- *Submit*: Based on the question or task raised by the entity in the storytelling, the user can submit his/her answer.
- *Change entity property*: The user alters the property of an entity in the storytelling. For example, the user kills a monster (i.e. change monster's life value).

3.4.5 Interactive Storytelling Methodology Overview

CHAPTER 3. AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

Figure 3.2: A Full Picture of Agent-Based Methodology for Interactive Storytelling

CHAPTER 3. AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

As shown in Figure 3.2, an agent-based interactive storytelling methodology is composed of the following models and components:

- (1) *Text-based Stories*: It denotes the designed text-based story scenario by the scriptwriter.
- (2) *Story parser*: The component retrieves the interested concepts from the story scenario. For the characters, the concepts include the characters' emotions (sorrow, happiness etc.), behaviors (walking, sleeping, resting etc.) and characteristics (age, gender, personality etc.). For the story contexts, the concepts include the background story, time, place, objects, and world construction etc. It provides the basis for the context modeling and the character modeling.
- (3) *Context Modeler*: It models the real-time simulation of the context variables which are retrieved from the game story scenario. It aims to build an immersive eco-system that the virtual agents inhabit. According to the scenario, some context variables might affect the interactions of the player and the virtual character in real-time. For example, when the weather in the game changes from sunny to rainy, the virtual character might need to find an umbrella before going out, and some of the character's activities (e.g. going to picnic) cannot be executed.
- (4) *Plot Modeler*: It constructs the game story plot from the story scenario. Then the plot will be assigned to the characters to perform individually.
- (5) *Character Modeler*: Each character modeler models and simulates a virtual character in the real-time. It includes the emotion modeling and the behavior modeling. Both the emotions and the behaviors of characters can change in real-time in correspondence to the stimulus. Take one scene of the story "*Little Red Riding Hood*" as an example, little red riding hood goes to grandma's house. The closer she is getting to the house, the happier she becomes. However, when the wolf appears, she becomes scared. When the wolf gets closer, little red riding hood runs away.

- (6) *S-MADE*: It is the toolkit to create all the agents (including director agent and character agents), and to load the goals into the agents. It will be discussed in details in Chapter 6.
- (7) *Interactive Storytelling System*: It is the final interactive storytelling system based on the initial text-based stories. The users are able to interact with the story director or story characters through the interactive storytelling system, in order to achieve a dynamic or personalized storytelling.

3.5 Interactive Storytelling Process

The process of interactive storytelling is the process of pursuing goals of the following agents: scriptwriter, director, and characters. A sample process is shown in Figure 3.3, which includes four steps: story authoring, story executing, scene dispatching, and character performing.

3.5.1 Story Authoring

Story authoring, or story generation is the process of generating the story plot, and defining the incidents. The scriptwriter creates the story plot that contains multiple storylines. Figure 3.3(a) shows the plot for a sample story scenario, which contains seven possible scenes from $S0$ to $S7$. The story scenario contains four alternative meaningful storylines as

$$S0 \Rightarrow S1 \Rightarrow S2 \Rightarrow S3 \Rightarrow S6 \Rightarrow S7 \quad (3.1)$$

$$S0 \Rightarrow S1 \Rightarrow S2 \Rightarrow S5 \Rightarrow S6 \Rightarrow S7 \quad (3.2)$$

$$S0 \Rightarrow S1 \Rightarrow S4 \Rightarrow S3 \Rightarrow S6 \Rightarrow S7 \quad (3.3)$$

$$S0 \Rightarrow S1 \Rightarrow S4 \Rightarrow S5 \Rightarrow S6 \Rightarrow S7 \quad (3.4)$$

The total number of storylines is calculated by the products of choices in the scenario. After preparing the plot, the scriptwriter agent sends the whole story plot to the director agent for execution.

Figure 3.3: Interactive Storytelling Process: Agents vs. Goals. (a) The Scriptwriter Agent Designs Story Plot with Various Choices; (b) The Director Agent Selects Story Scenes Dynamically; (c) The Story Scene S2 Involves Performance between Two Actors; (d) The Director Agent Dispatches the Scene to Two Actors as Their Behaviors.

3.5.2 Story Executing/Scene Selection

Although the interactive storytelling plot is non-linear, the director agent selects only one storyline dynamically based on the user interactions and context criteria. Scene selection is done based on fuzzy cognitive reasoning mechanism, which will be explained in detail in the next chapter. As shown in Figure 3.3(b), path 3.2 is selected in real time, in which scenes $S2$ is selected after scene $S1$ rather than $S4$, and $S5$ is selected after scene $S2$ rather than $S3$.

3.5.3 Scene Dispatching

Interactive storytelling in a virtual environment is similar to a drama, in which virtual actors eventually perform the scene rather than the storyteller narrating alone. Each scene of the plot might involve different actors acting together with assigned behaviors, so the director agent needs to distribute the scene to relevant actors dynamically. In the example, scene $S2$ involves communication between two actors as shown in Figure 3.3(c), which contains two concurrent behaviors belonging to two actors respectively. Therefore, the director agent separates the scene to two sub-scenes, assigns sub-scene $S2_1$ and sub-scene $S2_2$ to two actors respectively. Synchronization is used to synchronize the communications of actors and user interactions. The story plot modeling and task dispatching process are described in details in the next chapter.

3.5.4 Character Performing

After the actors receive assigned behaviors from the director agent, they will perform the acts in the sequence of temporal relationship, with tight synchronization. The composite behavior shows the abstract behavior, which can be further decomposed to atomic behaviors for direct execution. The actor agents are able to perform the behaviors with their own personalities and are affected by user interactions as well. Besides the behaviors, emotions and gestures are also very important to make the actors more believable, but they are still required to be consistent with the personalities and related behaviors.

3.6 Steps to Create Interactive Storytelling

The developers can create a live interactive storytelling from a text-based story through the following steps:

- (1) Identify the agents in the story
- (2) Identify the goals and actions for each agent and the conditions to handle user interactions
- (3) Design the goal net for each agent with Goal Net Designer
- (4) Run the agents' Goal Net with S-MADE Runtime

We would like to demonstrate the methodology step by step, with the example of the tale “Little red riding hood”.

3.6.1 Identify the Character Agents from the Story

In order to construct the multi-agent system for interactive storytelling, we need to figure out all the agents from the story script at the first stage. As shown in Figure 3.1, two agents are a must for every interactive storytelling. The first is the director agent, which is also called drama manager by some researchers. Its main responsibilities are

- Schedule the behaviors of characters
- Ensure the behaviors of characters are coherent with current story plot
- Handle the user interactions to select storyline

The other agent is the user agent, which is a representation of the player in the user environment. User profiling is very important in order to provide personalized storytelling. Besides the two kinds of agents, the characters which participate in the storytelling and are represented by names in the story script should be figured out. They will be modeled as character agents. “Little red riding hood” story is

“Little red riding hood is a happy girl. One day, mum tells her that grandma living in the forest is sick, so she asks little red riding hood to take some cakes to visit grandma. Mum warns her not to talk to strangers on the road. Little red riding hood prepares the cake and enters into the forest, where she meets the wolf. Wolf asks where little red riding hood goes. Little red riding hood tells the wolf that she is going to visit her sick grandma. The wolf finds the grandma first and eats her. It pretends to be the grandma and waits for little red riding hood. Little red riding hood finds that the one lying on the bed is the wolf, then she finds the hunter. The hunter kills the wolf and saves the grandma.”

The main characters include “Little red riding hood”, “Wolf”, “Mum”, “Grandma”, and “Hunter”. Then we need to define the belief, goals, as well as the actions of the agents, which can be extracted from the script as well.

3.6.2 Identify the Goals and Actions for Each Agent and Conditions to Handle User Interactions

We need to define the goals and actions for director agent as well as character agents separately.

The goal of director agent is to present the storyline. Due to the interactions from the user, the storyline is non-linear. In the “Little Red Riding Hood”, some goals of the director agent could be “prepare to visit grandma”, “meet wolf”, “talk to wolf”, “wolf eats grandma”, “detect false grandma”, “escape from the wolf”, “find the hunter”, and “kill the wolf”. As the current story is a linear narrative, the current goals of director agent will achieve a linear storytelling. In order to provide user interactions, we need to derive the related goals which might not appear in the story. There are two ways:

- (1) Find different versions of the storytelling; extract the same goals and different goals. For example, there are several versions of “little red riding hood”. In one version, little red riding hood fails to find the wolf lying in the bed, and she is eaten by the wolf eventually. Therefore, another goal “did not find wolf” can be an alternative to “find wolf”.
- (2) In order to convert a story which has only a linear version into interactive storytelling, we need to derive new goals and the conditions that might cause the new goals. For

example, we can consider the player who acts as little red riding hood doesn't meet the wolf on the road. Then we can have "little red riding hood meets grandma" goal.

The goal of each character agent focuses on how the agent performs in the virtual world itself as well as with others.

As designing the goals of director agent, the alternative goals of character agents could be designed in response to different user interactions.

3.6.3 Design Goals and Actions with Fuzzy Cognitive Goal Net Designer

Based on the abstracted goals and actions of the agents, we can draw the goal net with goal net designer and store them into the database for real-time execution. The director agent and character agents represent two levels of story executions. Therefore, the goals for director and characters are designed separately.

3.6.3.1 Design Goals for Director Agent

The story plot to be presented (i.e. the goal of the director agent) is modeled as shown in Figure 3.4, in which the original story is presented. After adding the alternative scenes (goals of director agent), an interactive storytelling scenario is shown as 3.5, in which there are multiple endings of the story. For example, the little red riding hood does not find out that the "grandma" on the bed is the wolf. The wolf eats the little red riding hood. On the other hand, the little red riding hood might not meet the wolf on the way, and goes to grandma's house successfully.

The director agent selects a storyline dynamically based on the user interactions and the state of context. A sample storyline of "Little red riding hood" in real-time is that, the little red riding hood fails to recognize the "grandma" and is eaten by the wolf. However, the hunter walks nearby and kills the wolf. The little red riding hood and the grandma are saved at last.

Figure 3.6 shows the details of the scene "detect the false grandma". In the scene, two char-

CHAPTER 3. AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

Figure 3.4: A Linear “Little Red Riding Hood” Story

Figure 3.5: A Non-linear “Little Red Riding Hood” Story with Alternative Goals of Director Agent

Figure 3.6: The “Detect the False Grandma” Scene in “Little Red Riding Hood” Story

acters “wolf” and “little red riding hood” act together in parallel. By using the dispatching algorithm, the “wolf” and “little red riding hood” get their respect goals. A sample Fuzzy Cognitive Map is created to simulate the context variables, interactions and goals in the scene as shown in Figure 3.7.

Figure 3.7: Fuzzy Cognitive Map in the Scene of “Detect False Grandma”

The goal that the agent will select depends on the events that affect the goals in real-time.

3.6.3.2 Design Goals for Each Character Agent

Director agent achieves its goal through guiding the characters agents achieve their goals. According to the scene dispatching mechanism, we can separate the goals for the wolf and little red riding hood who act in parallel. For example, the goal for the “Little red riding hood” agent can be seen in Figure 3.8. The goals of character agent will be loaded to the

Figure 3.8: Behaviors of Little Red Riding Hood in the Scene “Detect the False Grandma”

agent and run in real-time.

3.6.4 Running Characters’ Goal Net with S-MADE Runtime

After the goals of agents have been created, we can run the agents’ goals to execute the interactive storytelling. An interactive storytelling engine, Multi-agent development environment (S-MADE) will be illustrated in Chapter 6 in detail.

S-MADE runtime will load the goal net for agents automatically from the database. According to the user interaction and system dynamics, goals are loaded to the agents in real-time.

S-MADE runtime will load the goal for director first with the designed goal net ID. Based on the goal at a scene, it will load the relevant goal net for the character agents involved in the scene. Fuzzy Cognitive Map module runs at the background to handle user interactions and virtual dynamics, and provides the basis to load correct goals.

3.7 DIRACT

In order to simplify the agent methodology to generate interactive storytelling, the multi-agent system is further simplified and named as DIRACT, which is short for “Direct and Act”. It exhibits a flexible storytelling architecture which is composed of atomic DIRACT agents. It is a hybrid of the character-based interactive storytelling and the director-based interactive storytelling. DIRACT is an agent-based interactive storytelling architecture, such that the storytelling is composed as a multi-agent system. Each agent is called as DIRACT agent.

3.7.1 Structure of DIRACT Storytelling System

Agents are goal-oriented, autonomous objects, which work in specific context. The goal of a storyteller is to convey a story to audiences with certain interactions from the audiences. Interactive storytelling in the virtual environment involves agents with multiple roles, i.e., scriptwriter, director, and virtual characters.

Figure 3.9: Multi-agent Interactive Storytelling System of DIRACT

In DIRACT, there are no differences among the scriptwriter agent, director agent or character agents. Each agent is a DIRACT agent, which performs the role according to the

goals assigned in the storytelling. Therefore, the role of a DIRACT agent can be changed in real-time.

3.7.2 DIRACT Agent

Each DIRACT agent acts as both a director and a character. The properties of DIRACT agent include:

- *Atomic*: Each agent is atomic making it easy to plug in and out. This allows more customization in the storytelling process.
- *Inheritance*: The goal of an agent can be inherited from another agent.
- *Automate*: Each agent is a goal-oriented object to process the story element.
- *Robust*: If an agent does not perform well, it can be re-spawned by the storytelling system.
- *Adaptive*: The agent is able to adapt the context changes as well as user interactions.

The circles in Figure 3.9 show the agents involved in the interactive storytelling system. Each DIRACT agent can perform in the following roles:

Scriptwriter

A scriptwriter constructs the plot of the story. Different from the traditional story plot, the plot created by the scriptwriter agent is nonlinear, which provides choices and alternatives for audiences.

Director

A director is the storyteller, who is responsible for selecting and determining the storyline from the whole plot given by the scriptwriter. According to the user interactions U as well as context changes E , a reasonable storyline is chosen by the director dynamically.

Characters/Actors

The virtual characters are the virtual entities to perform the scenes of the story. AC_1 , AC_2 , and AC_3 are three sample actors. They take over tasks as behaviors from the director and perform in a temporal sequence. Among the actors, the one which is keen to convey the moral of the story is called protagonist. The actors interact with each other according to the plot, and the audience interactions U and context changes E may also affect the behaviors of the actors.

3.8 Methodology Evaluation

By reviewing the process of interactive storytelling, it is shown that, a plot-based approach focuses more on the first two steps, i.e., story authoring and story scene selection. On the other hand, a character-based approach focuses more on the last two steps, i.e., scene dispatching and actor performing; however even then, scene dispatching is normally done manually by the designers through the scripts, and not done automatically and dynamically. In the proposed hybrid system, user-awareness and context-awareness of the storytelling are achieved by the director agent and virtual actor agents through a fuzzy cognitive reasoning mechanism.

Feature comparisons with Mimesis, Character-based IS and Façade are shown in Figure 3.3.

3.9 Summary

In this chapter, we demonstrate an agent-oriented methodology for interactive storytelling (AOMIS), which combines the plot-based and character-based approached with goal-oriented agents. Through the methodology, story generation and story presentation are done in real-time; therefore, story plot coherence and character performance believability are both achieved.

In the interactive storytelling methodology, Fuzzy Cognitive Goal net is used as the plot model and character model, while Fuzzy Cognitive Map is used as the context model. Each

CHAPTER 3. AGENT-ORIENTED METHODOLOGY FOR INTERACTIVE STORYTELLING (AOMIS)

	Mimesis	Character-based IS	Façade	AOMIS
Context				
· Scalability				✓
· Dynamic	✓	✓	✓	✓
· Uncertainty			✓	✓
Character				
· Goal directed behavior				✓
· Cognition		✓	✓	✓
· Interaction		✓	✓	✓
Plot				
· Coherence	✓		✓	✓
· Hybrid			✓	✓
· Deferred Loading				✓
Methodology				
· Easy to model	✓	✓		✓
· Toolkit				✓

Table 3.3: Feature Comparisons with Mimesis, Character-based IS and Façade

agent is goal oriented and adaptive to user interactions and context changes. In order to model the goals and actions of agents efficiently, Fuzzy Cognitive Goal Net (FCGN) is illustrated in Chapter 4. Moreover, agents need to reason about user interactions and context changes in real-time, therefore, Evolutionary Fuzzy Cognitive Maps (E-FCM) which models both the probabilistic and fuzzy causal relationships among a set of real-time variables is proposed in Chapter 5.

CHAPTER 4

PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

4.1 Introduction

In the previous chapter, an agent-oriented methodology for interactive storytelling (AOMIS) is proposed to construct the interactive storytelling as a multi-agent system. In this methodology goal modeling of the agents is the key challenge, i.e., how to model the goals and behaviors of character agents and director agent.

Interactive storytelling in the virtual environment becomes very complex due to the dynamic or unexpected user interactions and context changes. A believable agent is goal-oriented and robust to environment changes. In order to model the complex situations in the interactive storytelling, an efficient goal model is required to model the agent. Goal net model is proposed by Shen [59] to model the goals of an intelligent agent in the hierarchical and temporal causal related manner. It can be used to model most software agent's goals. However, though it exhibits the temporal causal relationships among the goals, it lacks the ability to reason when presented with dynamic context changes and user interactions, which is important for agents in the interactive storytelling.

In this chapter, we propose an agent goal model, Fuzzy Cognitive Goal Net (FCGN), which empowers the generic goal net model with fuzzy cognition over events in dynamic changing environments. Moreover, we will illustrate how Fuzzy Cognitive Goal Net can be used in

the context of interactive storytelling, to model the goals of the director agent, character agents and other agents respectively.

4.2 Model Description

4.2.1 Basic Concept of Fuzzy Cognitive Goal Net

Agents are goal-oriented entities that plan their goals and actions automatically. In this chapter, Fuzzy Cognitive Goal Net is proposed as a novel goal-oriented modeling approach for conducting interactive storytelling in a virtual environment.

Fuzzy Cognitive Goal Net (FCGN), derived from Goal Net model, models goal, goal transition, and task in a hierarchical architecture of an agent, with fuzzy cognition over the relevant context variables. A sample Fuzzy Cognitive Goal Net is shown as Figure 4.1.

Figure 4.1: A Sample Fuzzy Cognitive Goal Net

A goal of an agent represents a target state that the agent intends to achieve in certain context. In order to model a story director and characters with complex interactions with the users, the director agent and the character agents need to achieve complex goals according to different interactions. Depending on the types of the interactions, different types of goals are assigned for each agent. A root goal is decomposed to a series of composite goals and atomic goals, while each composite goal is decomposed of a serious of atomic goals hierarchically.

4.2.2 Elements of Fuzzy Cognitive Goal Net

Fuzzy Cognitive Goal Net models the agent in two levels of automation: goal level and action level. Goal level means the agent is able to select its goal autonomously, while action level means the agent is able to select different action in order to pursue a typical goal.

The typical elements of a Fuzzy Cognitive Goal Net include *state*, *goal*, *transition*, *event*, *task*, which are defined as below.

Definition 4.6 *A state S is a piece of information of the agent within the current context. It can be represented as a tuple $[S_1, S_2, \dots, S_n]$, when S_n represents one attribute of the agent.*

The state of the agent depends of the context that the agent is in.

Definition 4.7 *A goal G is a target state that is pursued by an agent, which can be represented as a tuple $[P_c, N, A, R, T]$, where P_c is the pre-condition of the goal, N is the goal name, A represents the action list of the goal itself, R is the result after finishing the goal, and T is the time slice of the goal.*

As shown in Figure 4.1, a goal can be a *composite* goal or an *atomic* goal. An atomic goal means that the goal cannot be further divided into other goals, and it is represented with a circle. A composite goal is a goal that can be derived from other atomic goals or composite goals through the goal operations that is represented with a shaded circle as shown in Figure 4.1. Here, the divided goals are called *sub-goals* to the composite goal and the composite goal is called *super-goal* to the divided goals. A goal-oriented agent is able to pursue its goal automatically rather than to perform specific tasks only. Thus, a goal represents a state that an agent would like to achieve, which can be realized by the action of the agent or by other context changes. For example, an agent goal is “winning a champion title”, that can be achieved by 1) the agent beats other players; or 2) the other players fail to complete.

CHAPTER 4. PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

Definition 4.8 *A transition T is defined as a linkage leading from a starting goal to a ending goal, which contains a task list to be executed for the transition. It can be represented as a tuple $[G_s, G_e, N, A, T]$, where G_s is the starting goal of the transition, G_e is the ending goal of the transition, N is the transition name, A is the action list in the transition in order to achieve the ending goal G_e from the starting goal G_s , and T is the time slice for the transition.*

A transition links two related goals. A transition is a link from a starting goal to an ending goal. There are five types of transitions:

- *Direct*: The starting goal leads to the ending goal directly.
- *Choice*: One starting goal links to two or more ending goals. A selection variable is used to determine which ending goal is chosen after the starting goal.
- *Parallel*: One starting goal links to two or more ending goals which might be able to achieve at the same time.
- *Synchronization*: Two or more starting goals link to one ending goal. Each parallel transition will end with a synchronization transition.

Definition 4.9 *A task is an atomic action within a transition or a goal. It can be represented as a tuple $[Pre_C, N, T, Post_C]$, where Pre_C is the pre-condition of the task, N is the name of the task, T refers to the task function and $Post_c$ is the post-condition of the task.*

A task list is formed by a list of tasks. It can be called inside a goal or inside a transition. One agent fulfils its goals or transitions through executing a task list. In the task list, a task is an action executed by an agent. It can be a composite task or an atomic task. In the virtual environment, these tasks of an agent show how they act and interact with the players.

The typical atomic tasks for an agent in the virtual environment are shown as Table 4.1.

CHAPTER 4. PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

Figure 4.2: Transition Types of Fuzzy Cognitive Goal Net: (a) Direct; (b) Parallel; (c) Choice; (d) Synchronization

Type	Task
animation	idle
	walk
	run
	jump
emotion	happy
	cool
	angry
	sleepy
	confused
	sad
action	turn
	move
	teleport
sensory	get(%attribute)
communication	speak
	send
	pass

Table 4.1: Task Pool

CHAPTER 4. PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

Based on the atomic tasks, a composite task can be constructed. For example, a “celebrate” can be viewed as

$$celebrate(\%target) \leftarrow move(\%target), speak, happy$$

4.2.3 BDI View of Fuzzy Cognitive Goal Net

BDI is a well-known agent architecture that describes three main important features of an agent, i.e., belief, desire and intension [56]. Here,

- *Belief* is the prior knowledge of the agent. In the interactive storytelling, belief could be context variables, user preferences that agents are interested.
- *Desire* is the goal of an agent that it would like to pursue. In the interactive storytelling, the goal of an agent is to organize/author a story scene or perform a story scene.
- *Intension* is the action that can be performed by an agent.

In Fuzzy Cognitive Goal Net, belief as the context, is modeled in the cognitive network to provide real-time cognition; desire is the goal of an agent, which is modeled in a hierarchical structure; and intention is modeled as the tasks which can be executed by the agent.

4.2.4 Goal Operation

A composite goal means that the intended state is not simple to achieve based on the current state of the agent. However, it can be generated from the atomic goals with the goal operations. By using the goal operation, it is helpful to derive the new goal easily and keep the goal pool small (i.e. there is no need to keep all the goals of agents but only the atomic goals). As shown in Figure 4.3, there are different types of goal operations defined for Fuzzy Cognitive Goal Net:

- **Goal Intersection Operation $\cap$**

$$\mathbb{G} \leftarrow G_i \cap G_j$$

Figure 4.3: Goal Operation of Goals G_i and G_j (a) Union; (b) Intersection; (c) Not; (d) Following-by

The goal $\mathbb{G}$ is achieved when both the goal G_i and G_j are achieved. A goal can be the goal intersection of multiple goals, such as $G_1 \cap G_2 \cap \dots \cap G_n$.

- **Goal Union Operation $\cup$**

$$\mathbb{G} \leftarrow G_i \cup G_j$$

The goal $\mathbb{G}$ is achieved when either the goal G_i or G_j is achieved. A goal can be the goal union of multiple goals, such as $G_1 \cup G_2 \cup \dots \cup G_n$.

- **Goal Not Operation $\neg$**

$$\mathbb{G} \leftarrow \neg G_i$$

It means that the goal $\mathbb{G}$ is achieved while G_i is not achieved. Here, $\neg(\neg G_i) = G_i$.

- **Goal Following-by Operation $\Rightarrow$**

$$\mathbb{G} \leftarrow G_i \Rightarrow G_j$$

It means that goal $\mathbb{G}$ is achieved by achieving the goal G_i followed by achieving the goal G_j . The operation can be used to link a series of temporal causal related goals, such as $G_1 \Rightarrow G_2 \Rightarrow \dots \Rightarrow G_n$.

As a result, any complex goal can be represented as a set of atomic goals applied with the goal operations. For example, one goal of an agent is “to win the game”, thus it can be

represented as

$$G1 \leftarrow G2 \bigcup G3 \quad (4.1)$$

$$\leftarrow (G4 \Rightarrow G5) \bigcup (G6 \Rightarrow G7) \quad (4.2)$$

$$\leftarrow ((G8 \bigcap G9) \Rightarrow G5) \bigcup ((G8 \bigcap G10) \Rightarrow G7) \quad (4.3)$$

where, G2 means “defeat all the enemies”; G3 means “destroy enemies’ buildings”; G4 means “find an enemy”; G5 means “defeat the enemy”; G6 means “find enemy building”; G7 means “destroy the building”; G8 means “explore the map”; G9 means “examine the non-player character”; and G10 means “examine the buildings”.

4.2.5 Goal Evaluation

In order to evaluate the performance of an agent, there is a need to evaluate the goals of agent systematically. The goal evaluation is based on the following criteria:

- *Achievement*: it represents the benefit of achieving a goal.
- *Cost*: it represents the total cost in order to achieve the goal.
- *Gain*: it represents the difference between the achievement and the cost.
- *Completeness*: it represents whether the goal is fully achieved or partially achieved.
- *Distance*: it represents the level of deepest atomic goals under a root goal. The distance of an atomic goal is 1.
- *Complexity*: it represents the level of complexity in terms of goal steps and branches, which might be important in the interactive storytelling to find the shortest or optimal path to achieve a goal. It can represented as,

$$C = \prod_{s_r} (b_r)$$

here b_r is number of branches at each sub-goal and s_r is the number of steps from the start state to the end state.

4.2.6 Running Algorithm

4.2.6.1 Goal Execution

A FCGN agent pursues its goals in real-time with reasonable actions according to its cognition over user interactions and context changes and reasoning. The pseudocode of running fuzzy cognitive goal net is shown as below. Goals and actions are pushed into and executed in parallel queues, so that the agent can reach both the goal autonomy and action autonomy in real-time.

Procedure. 1 *RunningAlgorithmofFuzzyCognitiveGoalNet*

Require: Root Goal G

```

1: Push  $G$  into Goal Queue  $Q$ 
2: while  $Q$  is not empty do
3:   Pop goal  $g$  from  $Q$ 
4:   Percept Environment  $e$ 
5:   if  $g$  requires  $e$  then
6:     if  $g$  is Atomic then
7:       {it is an atomic goal}
8:       Get action  $A$  from  $g$ 
9:       Execute action  $A$ 
10:      Get next transition  $t$ 
11:      Get next goal  $g_n$  after  $t$ 
12:      Push goal  $g_n$  into Goal Queue  $Q$ 
13:     else
14:       {it is a composite goal}
15:       Get Sub-goals  $g_1, g_2, \dots$ 
16:       Push Sub-goals  $g_1, g_2, \dots$  into Goal Queue  $Q$ 
17:     end if
18:   end if
19: end while

```

4.2.6.2 Goal Dispatching

Fuzzy Cognitive Goal Net is used not only as the story plot modeling tool, but also as the modeling tool for character behaviors. This connects director's goal and characters' goals seamlessly. Therefore, story generation autonomy as well as character performing autonomy can be achieved simultaneously through goal decomposing algorithm.

The goal decomposing algorithm is used for act/behavior dispatching. It is an important

CHAPTER 4. PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

step wherein the director agent assigns behaviors to different characters in a scene autonomously and dynamically. It is done by splitting its goals into subgoals for characters to act in parallel; the interactions are represented as synchronize transitions. The pseudo code of goal decomposing algorithm is shown below.

Procedure. 2 *DispatchingtheScene*

Require: Scene $S = [AC_1, \dots, AC_n, B, C]$

```

1: if  $S$  is Composite then
2:    $Q \leftarrow Child(S)$ 
3: end if
4: while  $Q$  is not empty do
5:    $S_t \leftarrow Q$ 
6:   for  $i = 1$  to  $n$  do
7:      $B_i = \text{Find}(S_t, B, AC_i, C)$ 
8:      $AC_i \leftarrow B_i$ 
9:   end for
10: end while
11: Function:  $B_i = \text{Find}(S_t, B, AC_i, C)$ 
12: for  $AC_i$  IN  $S$  do
13:   if  $B_i$  is independent %parallel then
14:     RETURN  $B_i$ 
15:   else if  $B_i$  is dependent of  $B_j$  then
16:     if  $B_j$  is done then
17:       RETURN  $B_i$ 
18:     else
19:       RETURN IDLE {wait for synchronization}
20:     end if
21:   else
22:     RETURN IDLE
23:   end if
24: end for

```

An example is illustrated in Figure 3.3(c) and 3.3(d). The scene is separated into subgoals according to the actors, and effects of context on actors. After that, the subgoals are assigned to the actors as their behaviors. Concurrency and synchronization transitions are used here to model the independent goals of the actors and interactions among them. Currently, the goal decomposing algorithm is partially done, and has constraints of the character behaviors in the scene.

4.2.6.3 Fuzzy Cognition of Variables

In order to perform the goal selection and action selection, there is a need to model the selection criteria dynamically. In our approach, Fuzzy Cognitive Map is used initially. However, Fuzzy Cognitive Map is limited due to the pre-defined causal relationships among variables. Therefore, Evolutionary Fuzzy Cognitive Map (E-FCM) is proposed in next chapter to empower reasoning capabilities in the modeling, depending on the activation of the goal or transition, while the activation is done by some event or variable or prerequisite. Evolutionary Fuzzy Cognitive Map is used as the tool to simulate the events and variables in real-time, in order to provide a real-time basis for goal selection and action selection.

4.2.6.4 Goal Selection and Action Selection

Fuzzy Cognitive Goal Net provides two levels of selection mechanisms for the agents: 1) selecting the goal by goal selection and 2) selecting the action by action selection.

Since the goal of an agent might vary in different situations, how to select a suitable goal is very important for the agents.

There are two types of goal selection mechanism:

(1) Benefit Maximization Algorithm

Each goal and transition has its own benefit and cost. For a set of candidate goals or transitions, a goal or transition is selected if total benefit is highest.

(2) Heuristic Selection

For a set of candidate goals or transitions, a goal or transition is selected if the precondition is fulfilled. The precondition is simulated in the Fuzzy Cognitive Map.

The first mechanism is mainly used for normal agent-oriented services, while the second on targets to real-time games or interactive storytelling which require very fast responsiveness. Action selection is the low-level selection mechanism of the agent to select an appropriate action in order to fulfill a goal. The action selection is also made based on the selection criteria which are simulated in Fuzzy Cognitive Map.

4.3 Plot Model of Interactive Storytelling

Plot model is used to model the story plot. As the story plot refers to the storyline directed by the director, the plot model shows the goal of the director agent. In this section, we focus on modeling the story plot with non-linear storylines by using fuzzy Cognitive goal net to model director agent's goal.

Story authoring is an important step in interactive storytelling. In this step, the director's goal is to plan the story plot well to fit audience interactions and context changes. Goal Net is an expressive and efficient tool to model the goal of the director.

The goal of the storyteller/director is to present each scene meaningfully to the audience in temporal order. Each scene/event S_i is represented as a goal, the causal relationship between scenes S_i and S_{i+1} is represented as transition T_i . The tasks for director agent to transit from scene S_i to S_{i+1} are represented in the task list at transition T_i . As shown

Figure 4.4: Fuzzy Cognitive Goal Net with Alternative Storylines

in Figure 4.4, a general storyline is encapsulated within Goal Net. Moreover, multiple storylines are generated for user interactions or context changes, the goal net shows different possible storylines with the relation of 'Choice'. Therefore, the director has a combination of ascension and climax to compose the story in the process of storytelling. For example,

a possible storyline could be one of the following:

$$Start \rightarrow Exposition \rightarrow Ascension1 \rightarrow Climax1 \rightarrow Ending \rightarrow End \quad (4.4)$$

$$Start \rightarrow Exposition \rightarrow Ascension1 \rightarrow Climax2 \rightarrow Ending \rightarrow End \quad (4.5)$$

$$Start \rightarrow Exposition \rightarrow Ascension2 \rightarrow Climax1 \rightarrow Ending \rightarrow End \quad (4.6)$$

$$Start \rightarrow Exposition \rightarrow Ascension2 \rightarrow Climax2 \rightarrow Ending \rightarrow End \quad (4.7)$$

The *goals* in the goal net represent the story scenes. A complex scene is represented as a composite goal, and a simple scene is represented as an atomic goal. The composite goal can be further divided into atomic goals such that a complex scene is split into small scenes for story narrative.

The *transitions* of goal net are capable and adequate to describe different causal relationships between story scenes in interactive storytelling. The *sequence* transition is used when two scenes have direct temporal causal relationships. The *concurrency* transition is used when two scenes are independent, such that the presentation order by the drama manager is not important. The *choice* transition is the most important to user interaction and context variable changes, as it can lead to different scenes after the current scene. The *synchronization* transition is needed for the concurrent scenes, such that the next scene can be achieved only after the concurrent scenes are achieved. It is important to synchronize characters' behaviors for the interactions among them. Moreover, the combination of *transitions* allows the modeling of complicated relationships among scenes. Thus it is able to model a very complex storytelling.

The director agent is responsible for presenting the story scenes from the goal net according to causal relationships among the scenes. If the story is sophisticated, i.e., the Goal Net structure is complex, sub director agents might be summoned to present the sub scenes. In other words, a complex Goal Net is decomposed into several Goal Nets, i.e., sub Goal Net.

4.3.1 Fuzzy Cognition in Interactive Storytelling

Dynamic storyline selection and behavior selection are achieved by goal selection and action selection in the goal nets of the director and character agents respectively. The reasoning

CHAPTER 4. PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

mechanism is the basis to make such selections.

Goal Net does not specify a particular reasoning mechanism for an agent's action selection and goal selection. Instead, goal net designer has the flexibility to choose reasoning mechanism on specific scenario context.

Traditionally, rule-based or predicate reasoning techniques are the most commonly used approaches for agent reasoning, whereby the agents make the decisions based on the rules and context changes. However, it has been noted that predicate reasoning is not powerful enough to model the situation in which there are complex and dynamic relationships among the factors. Rule-based approach is simple and straightforward, but it is error-prone when handling complex causal relationships and not workable when input information is incomplete. Machine learning techniques such as neural networks and genetic algorithms are also used for reasoning purpose in some research work.

In the proposed intelligent storytelling model, Fuzzy Cognitive Map is able to model the dynamic evolving context changes as well as user interactions. This provides the basis for goal selection and action selection. The user interactions, context variables, and the decisions are encapsulated as related concepts. The causal relationships among different concepts are determined according to the expert knowledge or learnt from a knowledge base. Large weights are assigned to those more assertive causal relationships, and small weights are assigned to the less assertive causal relationships. As opposed to other reasoning mechanism (e.g. rule-based engine), Fuzzy Cognitive Map can be seen as a collection of rules such that it is not only concerned about the relationships between the causes and effects as a normal rule-based engine does, but also considers the relationships among the causes. Therefore, it provides a stronger reasoning ability than rule-based reasoning. It is robust to model complex relationships among various concepts. Moreover, it is able to work with incomplete information, thus it is helpful to increase or decrease the number of factors in real-time, which is hard to achieve for rule-based reasoning.

Besides the scene selection by the director agent, the virtual actor agents can also use FCM to do the behavior/task selection in real-time, to respond to user interactions and context changes, so believable and dynamic behavior for virtual characters can be achieved.

4.4 Character Model in Interactive Storytelling

In this section, we focus on using fuzzy Cognitive goal net to implement the character behaviors to generate the story.

In virtual storytelling, the virtual characters act the story scenes rather than the director narrates the story. Besides representing the goals for the director (i.e. plot), fuzzy cognitive goal net is also used to model the goals of characters, i.e., to model the act that characters need to perform. In interactive storytelling, neither the act for a single character nor the story plot is linear, since user interactions and context changes might also affect character performance. Unlike the director, character agents are only concerned about local performance, rather than plot for entire story. The goal of each character agent is to perform the behavior assigned/arranged by the director.

Each goal represents a target state of character that the character expects to achieve; each transition shows necessary behaviors/acts that the character needs to perform in order to transit from input state to output state.

As shown in Figure 4.5, the little red riding hood is modeled with a Fuzzy Cognitive Goal Net in a scene. After the little red riding hood enters grandma's house, she can choose

Figure 4.5: Fuzzy Cognitive Goal Net with Alternative Behaviors of Little Red Riding Hood

to ask “grandma, why is your mouth so long?” or “grandma, why are your hands so strange?”, depending on the real-time context “grandma is sleeping”, “grandma holds little red riding hood’s hand”, and “grandma speaks to little red riding hood”. The real-time context variables are simulated with Fuzzy Cognitive Maps.

4.5 Hybrid Interactive Storytelling by Combining Plot Model and Character Model

In this section, we focus on using fuzzy Cognitive goal net to implement a hybrid interactive storytelling with both director agent and character agents.

It is a great challenge to model the temporal structure of the story plot and user interactions seamlessly. As a goal model, Fuzzy Cognitive Goal Net (FCGN) can be used for interactive story authoring and character behavior modeling. The model is made up of two parts: a goal modeling tool based on Goal Net and the reasoning mechanism using Fuzzy Cognitive Maps (FCMs). Goal Net is used as the story plot representation for the scriptwriter and director agent and behavior representation for the character agents respectively. Fuzzy Cognitive Maps equip the director with the ability to create dynamic path by reasoning about user interactions and environment context, as well as strengthen a character's real-time task selection ability.

A scene of story/drama is regarded as a goal to be pursued by the director agent. The goals are loaded by the director agent according to the temporal relationships between the scenes in real-time. For a complex scene in the presentation path, it can be decomposed recursively into specific sub goals. Depending on the user interactions and context, different consequent goals may be reached after a certain goal, i.e., different scenes are achieved in different situations after the current scene. Fuzzy Cognitive Maps are used to make goal selection or task selection by analyzing the causal relationships among relevant concepts. The two modeling tools (i.e. Goal Net and Fuzzy Cognitive Maps) are illustrated in details in the following sections separately according to their functionalities in the interactive storytelling model.

As an extension to generic Goal Net model [60], Fuzzy Cognitive Goal Net perceives the goal related variables and reasons to make suitable goals. With the "choice" transition, different goals can be achieved based on user preferences or real-time interactions. In our virtual learning environment, the agent can present different learning contents to different learners.

CHAPTER 4. PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

As shown in Figure 4.6, the little red riding hood’s plot and the behaviors of ”little red riding hood” and the wolf (in “wolf eats girl” scene) are modeled within the Fuzzy Cognitive Goal Net seamlessly.

Figure 4.6: Fuzzy Cognitive Goal Net with Plot and Behaviors of Little Red Riding Hood Story

4.6 Discussions

4.6.1 Comparisons to Other Models

As an extension to Goal Net model, Fuzzy Cognitive Goal Net models both the agent’s goals and actions with the following advantages:

- Fuzzy Cognitive Goal Net provides a better knowledge base of storytelling for agents in the interactive storytelling.
- New properties are defined for the agents. For example, the goals are time constrained which is critical in the real-time systems.
- Fuzzy cognition over user interactions and contexts is modeled in FCGN while Goal Net is lack of it.

CHAPTER 4. PLOT MODEL AND CHARACTER MODEL: FUZZY COGNITIVE GOAL NET (FCGN)

Thus, FCGN is more suitable to model the agent in a dynamic environment, while Goal Net mainly targets to conventional applications.

Hierarchical Task Network [25] has been proven to be a successful task planning mechanism. The task is organized with different levels of abstractions in hierarchical structure. In Cavazza's character-based interactive storytelling [14], HTN is used to model behaviors for virtual characters. In a dynamic scenario, fulfillment of the precondition of certain task will let the character perform the task.

Comparatively, Fuzzy Cognitive Goal Net has a similar hierarchical architecture to present different levels of abstraction. Particularly, Fuzzy Cognitive Goal Net shows the goal abstraction more than task abstraction. Therefore, Fuzzy Cognitive Goal Net supports not only behavior autonomy (by action selection) but also goal autonomy (by goal selection). In a short term, an agent is able to select and execute an appropriate action in order to achieve its next goal; in a long term, the agent can do a proper reasoning to select the appropriate goal path to achieve its final goal.

According to Shen [60], modeling the goals is a higher level modeling of an intelligent agent than modeling the tasks (actions) or states, which are in turn, ways to model the internal desire and intention of the agent. It guides agent on what to do rather than how to do. Tasks are activities that the agent needs to perform in order to transit from one goal to another. However, action selection might not be enough to model scenarios in which the agent changes its intentions. The change of goal might be shown as a task change, context changes, or other forms. In interactive storytelling, each scene is represented as a goal, and the transitions between the scenes are represented as the *transitions*. For example, in the fairy tale "*Little Red Riding Hood*", little red riding hood wants to deliver some food to her grandma. For this small part of story, the initial scene is that little red riding hood prepares the food, and the final scene is that the grandma has the food and is very happy. For the transition, it may have tasks like little red riding hood going to grandma's house to give the cake, or maybe the wolf has eaten the grandma (context change), then the little red riding hood needs to recognize the wolf and asks the hunter for help.

4.7 Summary

In this chapter, we illustrate Fuzzy Cognitive Goal Net (FCGN) model which is used as the goal model for agents presented in the last chapter. Fuzzy Cognitive Goal Net combines the goal modeling ability of goal net and the cognition ability of Fuzzy Cognitive Map. Goal Net models the goal in a hierarchical way and connects the sub-goals in the temporal causal relation order. It facilitates the abstraction of the goal, which enables both goal autonomy and task autonomy for the agent. The director agent represents a story plot with a goal net, and selects story scenes in real-time. By splitting the goals, the director agent assigns acts to the characters. The characters perform their own acts. Fuzzy Cognitive Map models complex causal relationships among various concepts in a dynamic environment, which provides the basis for goal selection and action selection.

Context model is extremely important in the interactive storytelling to handle the dynamic context changes. Fuzzy cognitive map is not enough to model the properties of real-time causal relationships of contexts, which might be chaotic, time-constrained. Therefore, we extend the generic fuzzy cognitive map to probabilistic fuzzy cognitive maps (E-FCM) as our context model to empower the real-time simulation, which will be illustrated in detail in next chapter.

CHAPTER 5

CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

5.1 Introduction

Immersive gameplay, one of the most important factors for a successful serious game, refers to a game experience in the virtual world which is close to real life. Therefore, creating an immersive game world is the first step to an immersive gameplay.

According to Zyda [79], in order to achieve an immersive gameplay experience, a cognitive game design approach should enable developers to create theories and methods for modeling and simulating computer characters, stories, and human emotions through affective computing. As the causal relationships represent the way by which the players experience the game world through their own inference, there is a need to create a cognitive model to represent the causal relationships.

A typical game world includes two key elements: characters (non-player characters and players) and the contexts. In order to model a character vividly, a large number of variables are required, e.g., physical parameters (strength, age etc.), emotions (happiness, sadness etc.), and behaviors (moving, speaking etc.). There are several important issues for real-time character modeling, as outlined below.

- How can a character's behaviors and emotions be updated according to the changes in contexts?

- How can a character's behaviors be consistent with his/her emotions?
- How does a character affect the environment/context through his/her behaviors?

The main concern of context modeling is how the values of the context variables are updated believably in real-time. For example, when the number of fish in the ocean increases, the number of sharks which feed on the fish will also grow. After some time, when the rate of fish being preyed on by shark exceeds the rate of fish reproduction, the number of fish will drop. This is a typical network of causal relationships in an eco-system. Due to the complex causal relationships among contexts, the contexts will be updated in a dynamic and stochastic manner. It is challenging to simulate these updates in a virtual environment, especially for large-scale serious games.

In order to solve the problems mentioned above, an effective cognitive computational model is needed to model the causalities among the variables. So far, many cognitive computational models of dynamic causal relationships have been developed. Some of them are qualitative models for analyzing the trend of the events, e.g., rule-based expert system [24] and Markov-decision processes (MDP) [23] etc. EMA is a famous computational model by Gratch based on the cognitive appraisal theory [31], which combines previous works to be a sound model. It uses causal interpretation to describe the causal relationships among the events.

For serious games in the virtual world, there is a need to model the concepts with precise values rather than symbolic reasoning. Therefore, quantitative models would be more useful. Cathexis is a distributed model by Velásquez [69] to model the emotions through activation functions; however, it only models some basic emotions and reactions. Fuzzy Cognitive Map (FCM) [38] by Kosko is one efficient inference engine to model such complex causal relationships in both qualitative and quantitative way easily. Kosko and Dickerson [21] also modeled the hunting process of sharks and fishes in a virtual world using FCM. In [45], the author used FCMs to successfully model the intentions/movements of a sheepdog and sheep in a virtual world. As a generic model, FCM relies on quite a number

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

of assumptions. For example, the activation values of concepts are updated simultaneously at the same rate, and the causalities among the concepts are always in effect. However, as these assumptions may not always hold, FCM is not powerful or robust enough to model a dynamic and evolving virtual world. In order to overcome these shortcomings, numerous extensions of FCM have been proposed. In particular, Miao *et al.* [51] developed Dynamic Causal Network to model the concepts quantitatively with time variables. Moreover, the causality between two variables is not only fuzzy, but also probabilistic rather than deterministic. Song *et al.* [63] proposed probabilistic events to model uncertain concepts. In order to describe the general logical operation (AND/OR) of rules, rule-based FCM is also proposed [13]. In addition, Evolutionary Multilayered Fuzzy Cognitive Maps [49] was also used as an inference tool for a real-time system with evolutionary strategy. However, the models above are mostly used as inference engines rather than real-time modeling and simulation. On the other hand, Bayesian Network models the probabilistic causalities among the concepts, but does not describe the fuzzy causalities quantitatively. To the best of our knowledge, there is still a lack of good solutions that meet the requirements of virtual world modeling, in particular the modeling of the characters and the contexts.

Based on FCM, we propose a computational model Evolutionary Fuzzy Cognitive Map (E-FCM). E-FCM redefines concepts and their causal relationships, and re-designs the process of how the values of the concepts are updated [11]. In E-FCM, each concept has its respective evolving time schedule (as different concepts update asynchronously) and has a small self-mutation probability to update the value randomly. The causality between two concepts is not only represented with a single fuzzy value; a conditional probability is also used to quantify the probability of the causality. Therefore, E-FCM presents both fuzzy causalities and probabilistic causalities, and allows asynchronous concept updates.

We use E-FCM to model the dynamic concepts and their causal relationships in the virtual world, which includes the modeling of the characters and the contexts. The values of the concepts are updated asynchronously according to the changes in the values of the concepts affecting them, and the changes are subject to the causal probabilities. As a result, the

virtual world becomes more dynamic with believable characters and contexts, consequently enhancing the players experience in the interactions.

5.2 Context Modeling in Interactive Virtual Story-telling

5.2.1 Context Representation

According to the functionalities of the components, there are two main tasks in developing a serious game: 1) To create a believable game world, i.e., to model the characters and contexts; 2) To plan the activities of the characters according to the game scenarios.

A game world has two main categories: characters and contexts. Therefore, game world modeling can be seen as a combination of character modeling and context modeling.

Character is one of the most important ingredients of a successful story. Dynamic and believable virtual characters are important for players to have an engaging experience. Two most important aspects of character modeling are emotion modeling and behavior modeling. Because the characters interact with the virtual environment in real-time, the behaviors and emotions of the virtual characters should evolve over time in different time schedules.

Take the simulation game *The Sims*¹ as an example, the variables to define an avatar includes:

- Emotion (happiness, sadness, boredom etc.)
- Behavior (eating, celebrating, dancing, chatting, sleeping, listening music, etc.)
- Property (tiredness, hunger, cleanness, wealth, etc.)

Besides the character variables, some context variables which affect the gameplay include:

- Time
- Friends Around

¹The Sims: <http://thesims2.ea.com>

- House Decoration
- Foods Available etc.

When more dynamic concepts are involved, the gameplay becomes more engaging. However, this could be a barrier to the devices with limited computing power. Therefore, there is a tradeoff between gameplay experience and scenario complexity. We can assign different priorities to the concepts in order to suit different levels of requirements.

For serious games in virtual environments, the modeling of the characters and contexts is becoming increasingly complex, with the following properties:

- (1) *Complex Causal Relationships*: Causal relationships are complex, which include the mutual causal relationships between the characters and environment as well as the causal relationships between the emotions and behaviors.
- (2) *Dynamic*: The emotions and behaviors of characters keep on changing as the game-play goes on. The characters need to respond rationally to the real-time simulation, by displaying the correct behaviors and emotions.
- (3) *Randomness*: The virtual characters should not act in a deterministic way. There should be some randomness of the variables.

In order to solve the problems, after the variables of interest (e.g. emotions, behaviors and contexts) are retrieved with concept retrieval component, E-FCM will be used to model the causal relationships among them, and start to run in real-time. This process will be shown in detail in the following sections.

5.3 Game World Modeling with Evolutionary Fuzzy Cognitive Map (E-FCM)

Based on conventional Fuzzy Cognitive Map (FCM), we have proposed Evolutionary Fuzzy Cognitive Map (E-FCM) to simulate real-time variable states, and used E-FCM to model

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

the dynamic and complex causal-related context variables [11].

In E-FCM, the concept states evolve in real-time, based on their internal mental states, external assignment, as well as external causalities. Moreover, the concepts update their internal mental states asynchronous with a small mutation probability.

E-FCM models every temporal state, which is named as *Evolving State* in the running process, as a collection of concept values.

Figure 5.1 shows an overview of the E-FCM structure. The bounding box indicates the system enclosure (E), which comprises all the concepts and causal-related information. *Clock* is the reference time for the concepts to update their values, which is not explicitly considered by conventional FCM. The graph model of E-FCM is constructed with two main components: *concepts* and *causal relationships*. The definitions of concepts and causal relationships, as well as how they are initiated, are illustrated in subsequent sections.

Figure 5.1: Structure Overview of E-FCM (E is the System Enclosure, Clock is the Reference Clock, S_i is the Concept and R_i is the Causal Relationship)

5.3.1 Concepts

Concepts represent the variables of interest in a real-time system. A concept is defined as a tuple of properties:

$$S = [S_V, T, P_s] \quad (5.1)$$

where S_V is the state value, T is the state evolving time schedule, and P_s is the state mutation probability.

5.3.1.1 State Value

S_V denotes the fuzzy value of the concept, which is the same as FCM. For simplicity, it uses a real value from $[-1, 1]$ or $[0, 1]$. As different concepts have different unit scales, a concept value is represented on a relative scale over the concept standard value. For a system with N variables, the evolving state is represented as a vector $\mathbf{S}_V$:

$$\mathbf{S}_V = \begin{pmatrix} s_v^1 \\ s_v^2 \\ \vdots \\ s_v^i \\ \vdots \\ s_v^n \end{pmatrix} \quad (5.2)$$

where s_v^i is the value of concept i .

To model the characters in serious games, the emotions and behaviors are the concepts which need to be retrieved from the scenario first. In the story “*Little Red Riding Hood*”, the concepts might include: emotions (“happiness”, “fear”, “hesitation”), behaviors (“walking”, “running”, “calling for help”, “collecting mushrooms”), and the related external contexts (“wolf nearby”, “grandma nearby”, “weather”, “daytime”).

The value of a concept is a qualitative description, which ranges from 0 to 1. Take the emotion “happiness” as an example, in the range of $[0, 1]$, its value conveys that the character is “not happy at all”, “somewhat happy” or “extremely happy”.

5.3.1.2 State Evolving Time Schedule

In real-time, different variables might have different evolving time schedules (i.e. update time). With a system reference clock, all variables can update their states according to their own evolving time schedules. For a system with N concepts, it is represented as a vector $\mathbf{T}$:

$$\mathbf{T} = \begin{pmatrix} t^1 \\ t^2 \\ \vdots \\ t^i \\ \vdots \\ t^n \end{pmatrix} \quad (5.3)$$

where t^i is the evolving time schedule of concept i .

In computer games, a “tick” is normally used as a time base to update events. In our model, the evolving time schedule is calculated as the number of “ticks”. A commonly used time interval for one “tick” is 1 second. For example, we can assume that the refreshing rate for the emotions is 2 “ticks”, i.e., they update every 2 seconds. The refreshing rate for the behavior “runaway” is 1 tick, as little red riding hood is very sensitive to the context changes and responds quite quickly (e.g. when the “wolf” is near).

5.3.1.3 State Mutation

Besides the causal effects from other variables, each concept will also alter its internal states randomly in real time. Different concepts might have different stabilities. The *stability* is modeled with a very small mutation probability, based on a uniform random number generator. If the probability is big, the system would become very unstable. For a system with N concepts, it is represented as a vector $\mathbf{P}_s$:

$$\mathbf{P}_s = \begin{pmatrix} p_s^1 \\ p_s^2 \\ \vdots \\ p_s^i \\ \vdots \\ p_s^n \end{pmatrix} \quad (5.4)$$

where p_s^i shows the self-mutation probability of the variable i .

In the simulation, the probability value used would be normally less than 0.1. The experiments show that the system becomes unstable when it is greater. To model the characters in the game, normally we put 0 for those variables that are stable in the scenario. For variables that might not be so stable, (e.g. emotions), we can set a small value to simulate it.

5.3.2 Causal Relationship

Causal relationship R represents the strength and probability of the causal effect from one concept to the other concept. The uncertainty of the system variables can be twofold: *fuzziness* and *randomness* [39]. It is defined as the following tuple:

$$R = [W, S, P_m] \quad (5.5)$$

where W is the weights of the fuzzy causal relationships, S is the signs of the fuzzy causal relationships and P_m is the probabilities of the causal relationships.

5.3.2.1 Fuzzy Causal Relationships

The causal relationship between two variables, i.e., how much one variable will affect the other variable, is fuzzy. Some fuzzy terms used are, e.g., “much” and “a little”. For a system with N variables, the fuzzy causal relationships of the system are represented as an $N \times N$ weight matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} w_{11} & \cdots & w_{1j} & \cdots & w_{1n} \\ w_{21} & \cdots & w_{2j} & \cdots & w_{2n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{i1} & \cdots & w_{ij} & \cdots & w_{in} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ w_{n1} & \cdots & w_{nj} & \cdots & w_{nn} \end{pmatrix} \quad (5.6)$$

where i is the index of the causal concept, j is the index of the consequence concept, and w_{ij} , in the range of $[0, 1]$, indicates the fuzzy weight of the casual relation under which the variable i affects the variable j . A higher value of w_{ij} implies a stronger causal relationship.

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

S in the definition of causal relationship shows whether the causal relationship is positive (+ve) or negative (-ve), representing whether the increase of causal concept will lead to the increase or decrease of consequence concept respectively. It is usually combined with W in the computation.

Normally in the practice, the weight w_{ij} is determined by expert knowledge. The expert knowledge is gained from one or many experts within the domain who have clear understanding of the causal relationships. Each expert will assign a fuzzy value ranging from “-1” to “1” to the causal relationship from one concept to the other. “-1” represents that the increase of causing-from concept will lead to the decrease of the causing-to concept. “0” represents that there is no causal relationships between the two concepts (i.e. they are independent to each other). “1” represents that the increase of causing-from concept will lead to the increase of the causing-to concept accordingly. The n by n matrix represents the causal relationships from n concepts to n concepts respectively. If there are multiple experts available, an average value is taken for each number of the matrix.

Moreover, if there are available training datasets for the variables, the causal weights can be computed as the statistical correlation of the input data (changes in the causal variables) and output data (changes in the consequence variable). More formally,

$$w_{ij} = \frac{Cov(i, j)}{\sqrt{var(i) \times var(j)}} \quad (5.7)$$

where $var(i)$ is the variance of the changes in variable i , $var(j)$ is the variance of the changes in variable j , and $Cov(i, j)$ is the co-variance of the changes in variable i and the changes in variable j .

For the purpose of modeling, it is important to determine the causal relationships between any pair of two variables. For example, an increase of “the wolf is nearer” causes a decrease of “little red riding hood being happy” greatly. An appropriate value of the causality from “the wolf is nearer” to “little red riding hood being happy” would be -0.8. By tuning the fuzzy weight, the impact might vary in the simulation accordingly.

5.3.2.2 Probabilistic Causal Relationships

The uncertainty of the causality is the conditional probability of one event over another event. Some terms are used to describe the causal probability, e.g., “always” and “often”. For a system with N variables, it is represented as an $N \times N$ matrix $\mathbf{P}_m$ ($\mathbf{P}_m$ denotes the mutual causal probability):

$$\mathbf{P}_m = \begin{pmatrix} p_{1 \rightarrow 1} & \cdots & p_{1 \rightarrow j} & \cdots & p_{1 \rightarrow n} \\ p_{2 \rightarrow 1} & \cdots & p_{2 \rightarrow j} & \cdots & p_{2 \rightarrow n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ p_{i \rightarrow 1} & \cdots & p_{i \rightarrow j} & \cdots & p_{i \rightarrow n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ p_{n \rightarrow 1} & \cdots & p_{n \rightarrow j} & \cdots & p_{n \rightarrow n} \end{pmatrix} \quad (5.8)$$

where i is the index of the causal concept, j is the index of the consequence concept, and $p_{i \rightarrow j}$ indicates the conditional probability of the relation under which the change in variable i causes the change in variable j . With prior knowledge, $p_{i \rightarrow j}$ can be calculated as P_{ij}/P_i statistically. Here, P_{ij} is the probability that both the causal concept i change and consequence concept j change happen and P_i is the probability that only the causal concept i change happens.

The causalities are not guaranteed to exist in the real world. For example, “The wolf near” might not decrease “little red riding hood being happy” all the time, but with a high likelihood. A causality probability 0.8 can be used to describe it. The higher the probability is, the more likely the causality will fire in the simulation.

5.3.3 Weight Adjustments with Interactive Evolutionary Computing

The causal weight and probability matrices are crucial in the modeling process of E-FCM. Expert knowledge is used to determine these matrices. However, due to the inevitable subjectivity in weight and probability assignment, these values may have small errors which accumulate over the iterations of inference. Therefore, the E-FCM may not always work as the expert predicted. Worse still, because of the complexity of the inference process, it is not easy to identify and correct these small errors. The problem is further aggravated in

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

our application because the intermediate inference results are important in order to create believable behaviors, which differs from traditional FCM inference where only the final result is needed. Hence, it is necessary to find a method to align the results of the E-FCM to our expectations by adjusting the weights and probabilities.

Towards this end, in this thesis we employ the technique of Interactive Evolutionary Computing (IEC), originating from Dawkins' work [18]. The objective functions in traditional genetic algorithm or evolutionary strategies are replaced by subjective selection of human users. Believability, in our opinion, is a subjective measurement. It is difficult to determine numerically and objectively that which one is more believable among a number of scenarios. To cope with this problem, we resort to the subjective judgments of the expert once again. With IEC, the expert not only designs the matrices, but also examines the results produced and selects those resembling the real world more closely than others. The selected candidate solutions are then mutated and recombined to produce the next pool of candidates. The reproduction-selection loop repeats until the result is deemed by the expert as satisfactory, or a maximum number of iterations are completed.

The flow diagram to adjust the weights with interactive evolutionary algorithm is shown in Figure 5.2. The process includes the following steps:

- (1) Initialize a single E-FCM by assigning the weights and probabilities according to expert knowledge.
- (2) Create a number of variations of the initial E-FCM by applying the mutation operator.
- (3) Use the crossover operator to create new E-FCMs. The new weights are weighted sum of weights in E-FCMs at Step 2.
- (4) Mutate the weights with a small value.
- (5) Run each E-FCM for N times, summarize the results and present those results to the expert.

Figure 5.2: Flow Diagram of Weight Adjustments with Interactive Evolutionary Computing

- (6) The expert selects several candidate E-FCMs as the parents of the next generation. If the running cycles are reasonable in the view of the expert, it is rated as “good”, otherwise it is rated as “bad”. Select the E-FCMs with top “good” rate.
- (7) If the “good” rate is higher than 90%, we consider the optimization process to be stable, and the output E-FCM is the optimized version of E-FCM in Step 1. Otherwise, continue the optimization process from Step 2.

5.3.4 Pseudo-code of Running E-FCM in Real-time

After the concepts and the causal relationships are confirmed, we are able to run the simulation with the E-FCM. Here is the pseudo-code for running the E-FCM in real-time.

Some terms as used in the pseudo-code are defined as follows.

- (1) $\mathbf{f}(\cdot)$: the activation function to regulate the state values, e.g., bipolar, tri-polar and

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

Procedure. 3 E-FCM(n)**Require:** Global clock t

-
- 1: Extract the relevant concepts into *concept_list*
 - 2: **for all** $Concept_i$ in *concept_list* **do**
 - 3: **if** $Concept_i$ at Update time, i.e. $t \% T_i = 0$ **then**
 - 4: Update the concept value as
 - 5: $\Delta S_i^{t+T} = f(k_1 \cdot \sum_{j=0}^n \Delta S_j^t \cdot w_{i,j} + k_2 \cdot \Delta S_i^t)$ //addition subject to probability
 - 6: $S_i^{t+T} = S_i^t + \Delta S_i^{t+T}$ //self mutation with probability
 - 7: **end if**
 - 8: $t \leftarrow t + 1$
 - 9: **end for**
-

logistics.

- (2) **Variable State S_i^t** : state value for the concept variable i at time t .
- (3) **Variable State Change ΔS_i^t** : state value change for concept variable i at time t .
- (4) **Evolving Time Schedule T** : time for concept i to update its value. Different concept may have different evolving time.
- (5) **Time Slice t_0** : an atomic time slice to update all the variables.
- (6) **k_1 and k_2** are two weight constants.

Here, the summation of $\Delta S_j^t \cdot w_{i,j}$ is subjected to the conditional probability P_m^j , and the summation of ΔS_i^t is subjected to the self-mutation probability P_s . The computation complexity of the algorithm is $O(n^2)$, where n is the number of concepts.

5.3.5 Game World Context Modeling Procedure with E-FCM

The procedure to model the game world with E-FCM as shown in Figure 5.3, is presented in details below.

- (1) Collect the relevant concepts (nouns or phases of descriptions) of game world, e.g.,
 - a) behaviors, emotions and attributes for characters and
 - b) context variables.

For example, a simple scenario is, “Little red riding hood goes to the grandma’s house. The sun rises, more mushrooms come out. Her happiness increases because

Figure 5.3: Game World Context Modeling Procedure with E-FCM

she is able to collect a lot of mushrooms.” Here, the concepts include “the altitude of the sun” and “mushroom growing”, “little red riding hood being happy”.

- (2) Pre-processing of the concept values: fuzzification or normalization, i.e., mapping the values into the range of $[0, 1]$ with respect to their respective standard values.

The concepts extracted from the game scenario use different units and different scales, e.g., “the altitude of the sun” (Concept A) can be expressed as the angle between the sun and the ground (with a value from 0 degree to 90 degrees), “mushrooms growing” (Concept B) is expressed as, “the number of mushrooms”, which takes a value from 10 to 100. In order to represent the causal relationships between the two concepts, we need to map both them to a value between 0 and 1. It can be done by the fuzzification or normalization process. For concept A, 0 degree is mapped to 0; 90 degrees is mapped to 1 and t_0 degrees is mapped to $t_0/90$.

- (3) Calculate the evolving time schedule for each variable and self-mutation probability. Select a “tick” time of T for the serious game, which is usually the minimum evolving time schedule of all the concepts in the game (e.g. 5 seconds). Compute the evolving

time schedule as t/T (t is the evolving time schedule in the game). If the expert observed that the evolving time schedule of a concept in the game is 10 seconds, the evolving time schedule used in the E-FCM would be $10/5 = 2$.

- (4) Derive the necessary causal relationships (verbs) and connect the relevant variables with directed arcs. The weight matrix is normally a sparse matrix. By reducing the unnecessary causal relationships, the calculations of the update process can be reduced.
- (5) Calculate the causal weights and probabilities (as shown in Section 3(C)).
- (6) Run the E-FCM to simulate the game world (as shown in Section 3(D)).
- (7) Convert the concept values back to the real values in the game engine, i.e., defuzzification, which is the reverse operation of Step 2.

5.3.6 Comparisons to Other Computational Models

Rule-based models are by far the most commonly used methods to model the causal relationships. However, nearly all the computational models use rules in different ways. The advantage of Rule-based models is that, they are easy to construct and straightforward for simple systems. However, rule-based systems also have some limitations. If the system is too complex, a large number of rules are required to be generated and the rules become very complex. Moreover, the predefined rules might not be complete, thus the rule-based system does not work if the real-time state is new or the information required by the rules is incomplete.

Comparatively, E-FCM is an implementation of rule-based model, which embeds the rules into a graphical representation, thus providing a straightforward way to represent the system for easy construction. An E-FCM model can be considered as a combination of FCM and Bayesian Network, as it models the causal relationships quantitatively with probabilities.

E-FCM is technically equivalent to FCM when the probabilistic causal relationships and the

evolving time schedules are not used. As FCM does not model the non-deterministic causal relationships, the same set of simulation outcomes are produced for each and every run of the gameplay. Virtual worlds modeled with FCM in [21] [45] are proven to be rational by the authors, but they are not adequate for serious games, as the players will lose interest if they experience the same world repeatedly. On the other hand, E-FCM is technically equivalent to Bayesian Network when the fuzzy causal relationships are not considered. Bayesian network models the randomness of the causal relationships through belief, but does not do well to simulate the concepts quantitatively. As often in the games, players are more engaged by actual numbers (e.g. health is 80 points) than the vague descriptions (e.g. health is low).

There are also some extensions to rule-based models to model the characters' emotions and behaviors. Among them, Fuzzy Logic Adaptive Model of Emotions (FLAME) is one well-established model by El-Nasr [23]. Similar to E-FCM, it involves fuzzy logic and probabilities in the computation. Fuzzy rules are used to compute the emotions and the behaviors in real-time, however, the generation of the fuzzy rules would be tedious at times as you need to consider the relationships among multiple causes and one consequence.

Comparatively, E-FCM is a generic model that is capable of modeling the causal relationships among the concepts in real-time systems. It can be used to model not only the characters, but also the contexts in the virtual world. Instead of a complex fuzzy rule which involves multiple causes and one consequence, it simply defines the causal relationship between any two concepts in the system with a clear graphical representation. As a result, it makes the construction easier and faster. Besides enabling fuzzy and probabilistic causal relationship, the concepts are updated in an asynchronous way, which is required in the virtual world modeling.

5.4 Case Study: Village of Mystery

In order to validate the E-FCM model, we illustrate this use with a case study based on a serious game for science learning, namely, the "Village of Mystery". Here is the story scenario:

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

“...The scientist John comes to a village for investigation of a mysterious disease. Villagers fall sick, but the disease has not been identified and no cure has been found. He needs to visit the village, talk to people around, do experiments and find some clues.”

The game is designed for educational purpose, aiming to help the students learn different kinds of diseases from the exploration in the virtual world.

5.4.1 Experiment Setup

In order to achieve a believable “John” with context awareness, the following causalities need to be modeled: how “John” is affected by the environment (e.g. water and mosquitoes), how the emotion of “John” changes as the story continues. To describe the elements in the gameplay, an E-FCM is constructed as shown in Figure 5.4. A total of nine concepts are

Figure 5.4: The E-FCM Model of the Sample Story Scenario (The Part Inside the Solid Line Shows the Character Model and the Part Outside Shows the Context Model. ‘+’ Means “Positive Causal Relationship” and ‘-’ Means “Negative Causal Relationship”).

extracted in the scenario, which are updated in real-time:

C_1 : Polluted Water (0 - Not polluted water; 1 - Very polluted water)

C_2 : Dirty (0 - Not dirty; 1 - Very dirty)

C_3 : Mosquito (0 - No mosquitoes; 1 - Lots of mosquitoes)

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

C_4 : Energy (0 - No energy; 1 - Full of energy)

C_5 : Walk Speed (0 - Stop walking; 1 - Walk fast)

C_6 : Happy (0 - Not happy; 1 - Very happy)

C_7 : Get Food (0 - Get no food; 1 - Get lots of food)

C_8 : Clean (0 - Clean nowhere; 1 - Clean everywhere)

C_9 : Health (0 - Totally sick; 1 - Very healthy)

Among the concepts, C_1 to C_3 are context variables that will affect the character's attributes and can be changed by the character. C_4 to C_9 are the character variables, including properties (energy and health), emotion (happy) and actions (walk, get food and clean).

The E-FCM can be constructed as two sub E-FCMs. The first E-FCM circled in Figure 5.4, represents the knowledge of the character; the second E-FCM, outside of the circle represents the knowledge of the contexts. The entire E-FCM model can be decomposed into four parts as follows:

- Internal of Character Model
- Internal of Contexts Model
- Characters' Actions over Contexts
- Contexts' Effects over Characters

Therefore, different domains of knowledge can be combined to suit more complex scenarios. This is an advantage inherited from FCM.

The weight matrix and the conditional probability matrix of the causal relationship are determined with an expert's knowledge as shown in Table 5.1 and 5.2 respectively.

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

W_{ij}		j								
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
i	C_1	0	0	0.6	0	-0.4	-0.4	0	0	-0.8
	C_2	0	0	0.4	0	0	-0.5	0	0	-0.8
	C_3	0	0	0	0	0	-0.8	0	0	-0.9
	C_4	0	0	0	0	0.8	0	0	0.8	0
	C_5	0	0	0	-0.7	0	0	0	0	0
	C_6	0	0	0	0	0.5	0	0	0.5	0.7
	C_7	0	0	0	0	0	0	0	0	1
	C_8	0	0	-0.9	-0.8	0	0	0	0	0
	C_9	0	0	0	0.5	0.8	0	0	0.6	0

Table 5.1: Weight Matrix of the Causal Relationship

$P_{i \rightarrow j}$		j								
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
i	C_1	0	0	0.8	0	0.2	0.4	0	0	0.9
	C_2	0	0	0.6	0	0	0.5	0	0	0.8
	C_3	0	0	0	0	0	0.5	0	0	0.8
	C_4	0	0	0	0	0.8	0	0	0.7	0
	C_5	0	0	0	0.4	0	0	0	0	0
	C_6	0	0	0	0	0.6	0	0	0.5	0.6
	C_7	0	0	0	0	0	0	0	0	0.9
	C_8	0	0	0.8	0.7	0	0	0	0	0
	C_9	0	0	0	0.3	0.8	0	0	0.6	0

Table 5.2: Conditional Probability Matrix of the Causal Relationship

Take t_0 as a unit of evolving time, the evolving time for the nine variables are calculated as

$$(1 \ 3 \ 1 \ 1 \ 2 \ 3 \ 2 \ 3 \ 1)$$

Here, we consider, the concepts “Dirty”, “Happy” and “Clean” are updated in every three cycles; the concepts “Walking Speed” and “Get Food” update for every two cycles; and the rest of the concepts are updated in every cycle. Suppose the initial state vector representing the concepts $C_1, \dots, C_9$ is encoded as

$$(1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1)$$

In the virtual world, it shows as, 1) the water is polluted; 2) there are dirties around; 3) there are no mosquitoes around; 4) John starts with full of energy and health, fast walking speed and happy mood; and 5) John does not clean any place and does not get food.

5.4.2 Experiment Results

5.4.2.1 Running results with E-FCM

Figures 5.5 (a)-(c) show the running cycles of the concepts “number of mosquitoes”, “health” and “energy” in three rounds of experiments with E-FCM modeling respectively. The running results of the three concepts are represented with a solid line, a dot dashed line and a dashed line respectively. There are no external assignments conducted over the concepts in the running process. As shown, the values of the concepts are updated asynchronously. Moreover, with the same initial vector, the running cycles are different in different rounds of experiments. This is due to the involvement of the probabilistic causalities as well as the small mutation probability of the concept values in the E-FCM model. In addition, the causalities among the concepts are shown to meet our expectations. For example, when the “number of mosquitoes” increases, the “health” of the character drops; on the other hand, when the “mosquito” decreases, the “health” of the character improves. The running activity patterns of “health” and “energy” also show the “causal increase” relationships between the two concepts. The lag time between the consequence concept

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

(a):

(b):

(c):

Figure 5.5: Results of Concepts “Mosquito”, “Health” and “Energy”: (a) First Round Experiment of E-FCM; (b) Second Round Experiment of E-FCM; (c) Third Round Experiment of E-FCM.

Figure 5.6: Results After Changing Value of “Clean” to 1 Following Simulation Shown in Figure 5.5(b)

and the causal concept depends on the evolving time schedules of the two concepts. As the evolving time schedules for the three concepts are all 1, the consequence concept will be updated one cycle after the causal concept, as observed from the figure.

5.4.2.2 Effect of changing a concept value

As shown in Figure (b), the “health” of the character drops to a very low level. According to the E-FCM in Figure 5.4, one way to increase the “health” would be “cleaning the area” to reduce the “mosquitoes”. Once the character in the virtual world assumes the behavior of “cleaning the area”, the variable “clean” has a positive change to the value of 1.

The running cycles after the change of the concept “clean” are shown in Figure 5.6. We can see that, the “health” of the character increases gradually. However, the “energy” of the character remains low. We thus conclude from the E-FCM that “cleaning the area” actually consumes the energy of the character greatly.

5.4.2.3 Effect of changing the evolving time schedule

As an important feature exhibited by E-FCM, the values of concepts are updated asynchronously. Figure 5.7 shows the running sequences when the evolving time schedule of the concept “health” changes from 1 to 5. It can be useful to model the scenarios, wherein different characters have different rates of health recovery.

Figure 5.7: Running Results After Changing Time Schedule of “Health” from 1 to 5.

Figure 5.8: Running Results of the Scenario with FCM

5.4.2.4 Running results with FCM

FCM, as a basis of E-FCM, provides a good foundation to model the causal relationships among a set of concepts quantitatively. To compare the simulation results of E-FCM with FCM, we model the variables with the same weight matrix and ignore the evolving time steps and the probabilistic causal relationship. The results of FCM model are depicted in Figure 5.8. Different from E-FCM model, all the states are updated in the same cycle. As shown in the figure, the variables reach the equilibrium loop quickly in around 5th cycle. The running cycles are the same in different rounds of experiments when the initial states are fixed. This shows a major limitation of FCM in modeling a real-time world, as a player will experience the same set of events every time he/she plays the game.

CHAPTER 5. CONTEXT MODEL: EVOLUTIONARY FUZZY COGNITIVE MAPS (E-FCM)

In comparison, E-FCM allows the concept states to cover most combinations of the entire state vector space, i.e., the player can experience the dynamics of a different virtual world each time.

5.4.2.5 Virtual World Implementation

In order to test the E-FCM model in the virtual world modeling, we implement the game world with the Torque game engine. Two screenshots are taken in the first round of the experiments as shown in Figure 5.9. The running sequence of the variables is shown in Figure 5.9 (b). Figure 5.9 (a) shows that, the character is in good health when he is far away from the dirty water initially; Figure 5.9 (b) shows that the health level drops when the avatar approaches the dirty water and the mosquitoes. As opposed to traditional rule-based modeling, the variables evolve even when no rules are executed, thus presenting a more dynamic picture of the virtual world. As a result, the character is more believable.

(a):

(b):

Figure 5.9: Screenshots at the First Round Experiment: (a) Avatar with Full Health; (b) Avatar with Decreased Health with Dirty Water and Mosquitoes Nearby

5.4.3 Evaluations

Evaluating the quality of user experience in a gameplay is a subjective matter. Based on Murray [53], user experience can be evaluated in terms of three key aspects, namely *immersion*, *agency*, and *transformation*.

Immersion is the feeling of involvement in the virtual environment, and the ability to interact with the environment. With the proposed model, the characters and the contexts are closely related once the causal relationships are defined. The affective computing increases the immersive experiences of the players.

Agency is the feeling that empowers the user to take actions in order to fulfill its intention. In our dynamic world, the player is able to carry out some actions over the contexts in order to achieve the goals. For example, as shown in last section, the player can increase its “health” by “cleaning the area”.

Transformation refers to the variety of the world presentation. Different players may experience the game world differently as E-FCM simulates the virtual world and characters dynamically and stochastically. For example, once the “mosquito” increases, the “health” of the character drops; after the player “cleans the area”, the “mosquito” decreases and the “health” of the character increases. The detailed evaluations based on the subjective metrics will be conducted in the future work.

5.4.4 Discussions

As an inference tool, FCM produces the desired inference results after the model stabilizes. More importantly, E-FCM shows the evolution of the states in real-time. This is essential as a simulation tool for modeling real-time characters and contexts.

Compared with FCM, E-FCM has the following improvements:

- (1) It allows a different update time schedule for each variable. For example, the value of “actions” can be changed faster than that of “emotions” in a gameplay.
- (2) It enables the self mutation of context variables, which presents the dynamics of the world variables as evolving behavior.
- (3) It involves the probabilistic causality among the variables, which reflects realistic relationships among the concepts, and adds more dynamics to the character as a result, i.e., the character will not act in a deterministic way.

Though E-FCM is similar to other extensions for FCM with the concepts of evolving strategy and probabilistic events, E-FCM models the entire process of emotion and behavior evolution as a simulation engine. Therefore, the evolution of the state vectors are the main concerns rather than the equilibrium state vectors, as each evolving time state shows a state of the system in real-time. This is important for describing a believable virtual environment, and for the intelligent agents to make decisions in real-time.

5.5 Summary

In this chapter, Evolutionary Fuzzy Cognitive Map (E-FCM) is proposed to model the dynamic variables of a virtual world and their causal relationships in serious games or interactive storytelling. Beyond the fuzzy causal relationships modeled in FCM, the probabilistic causal relationships among the variables are modeled, and the variables update their states with respect to individual time schedules. By modeling the causal relationships among the dynamic variables, characters and contexts are dynamic and believable, so as to ease the effort as presented, which provide the players a more engaging experience. In the future work, we shall explore the automatic methods for learning non-linear causal relationships, as represented by the weight matrix and the probability matrix, so as to ease the effort in model construction. E-FCM enhances the Fuzzy Cognitive Goal Net with a more realistic reasoning and simulation over the agent behaviors.

FCGN and E-FCM are two most important models in the agent-oriented methodology for interactive storytelling (AOMIS) to model the agent goals and story contexts. In next chapter, we will illustrate an interactive storytelling engine namely S-MADE to create an interactive storytelling system from scratch with agent creation and goal development.

CHAPTER 6

AGENT-BASED INTERACTIVE STORYTELLING ENGINE: S-MADE

6.1 Introduction

Currently, there are a lot of research focusing on the interactive storytelling models and methodologies. However, there is a lack of successful deployments of interactive storytelling methodology and toolkits. This barrier prevents educators or developers without sufficient programming skills from developing their own interactive storytelling from a story scenario. This is mainly due to the lack of effort between agent modeling design and the agent implementation. There are not many practical tools that aid the agent developers to implement agents directly. Some agent platforms are available to handle the normal computer questions (e.g. Java Agent Development Framework (JADE)¹), while they are not quite suitable for agent development in interactive virtual storytelling. In the area of interactive digital storytelling, successful products are rare. There are a few well-known tools, e.g., Alice² from Carnegie Mellon University, Storytelling Alice [37] which enables the non-programmers to create simple storytelling in the virtual world through the graphical user interface, Storytronics³ by Chris Crawford.

However, the current storytelling development tools exhibit the following limitations:

¹Java Agent Development Framework: <http://jade.tilab.com/>

²<http://www.alice.org>

³<http://www.storytron.com/>

- Difficult to develop non-linear storylines as well as to refine the character behaviors at the same time in the interactive storytelling.
- Limited to certain 3D virtual environments only, making it difficult for developers with different requirements and different backgrounds to extend the storytelling to their own 3D virtual environments.
- Hard to model complex causal relationships among user interactions and context variables that might cause changes in the story plot or character behaviors.
- Hard to reuse the storytelling scenarios developed previously. A new scenario is hard to extend from several previous story scenarios with minor tunings.

In order to facilitate the development of interactive storytelling in the virtual environment, an agent-based interactive storytelling engine, namely S-MADE, is proposed and developed based on our agent-oriented methodology for interactive storytelling (AOMIS). S-MADE stands for multi-agent development environment for interactive storytelling, which is a customized version of the generic multi-agent development environment (MADE). The engine provides all the facilities to create intelligent director agent and believable character agents with the guidance of story scenarios. Moreover, as S-MADE isn't tied to a fixed rendering engine as Storytelling Alice is, but provides a generic interface for agent implementation, thus different virtual environments can be used for deploying the interactive storytelling.

6.2 Multi-Agent Development Environment for Storytelling (S-MADE)

6.2.1 Background: Multi-Agent Development Environment

Multi-Agent Development Environment (MADE) is developed by Li and Shen [42] [78] to create and run goal-oriented agents which are modeled with Goal Net. It can create dummy agents and load the goal net into the agents and start to run. As a black-box design, after the goal net is created with the goal net designer, MADE will schedule all the goal selection and task selection.

6.2.2 S-MADE Components

Interactive virtual storytelling differs from other application in the agent domains, it requires a better understanding over the story and make meaning schedule over the characters' activities. In order to facilitate the interactive storytelling authoring, we have developed a Multi-agent Development Environment for Storytelling (S-MADE), which is a customized version of Multi-Agent Development Environment (MADE).

The overall architecture of S-MADE is shown in Figure 6.1. The S-MADE architecture

Figure 6.1: Overall Architecture of S-MADE

includes the following components:

- *Fuzzy Cognitive Goal Net Designer*: This component is used by the designer or scriptwriter to design the fuzzy cognitive goal net of an agent, including the goals, actions and selection conditions. Moreover, fuzzy cognitive map is developed to model the causal relationships among the variables (contexts, user interactions and selection condition etc) as well.

CHAPTER 6. AGENT-BASED INTERACTIVE STORYTELLING ENGINE: S-MADE

- *S-MADE Runtime*: The S-MADE runtime mainly has three functionalities: 1) it creates the dummy agents; 2) it loads the fuzzy cognitive goal net into the agents; and 3) it runs the agents inside the virtual environments.
- *Scheduler*: The scheduler schedules the executions of agent creator and fuzzy cognitive goal net loader.
- *Agent Creator*: The agent creator creates a dummy agent, assigns the new created agent with a null goal initially and places the new agent into the agent container.
- *Fuzzy Cognitive Goal Net Loader*: This component loads the fuzzy cognitive goal net from the database into the agent.
- *Agent Factory*: The agent factory is an agent container that builds and runs all the agents. Once an agent is created in the agent container, it will load the fuzzy cognitive goal net of an agent and start to run in real-time.
- *Rendering Engine*: Each rendering engine provides the 3D virtual environment in which the interactive storytelling is carried out. The user is able to login the virtual environment through the rendering engine and interact with the virtual world and virtual characters. Two rendering engines we have tested for deploying interactive storytelling include ActiveWorld by Renderware⁴ and Torque 3D Game Engine by Instant Actions⁵. The software provides the virtual world and also the virtual avatars for the story presentation. The system is constructed with a server-client paradigm. Hence, multiple users can share a common virtual environment, which enables the storyteller to present interactive storytelling to multiple users simultaneously.
- *Rendering Adapter*: As a high-level agent system, the running of S-MADE is independent of the virtual environments (i.e. rendering engine). As each rendering engine uses a different application programming interface (API), the current storytelling

⁴ActiveWorld: <http://www.activeworlds.com>⁵Torque 3D Game Engine: <http://www.garagegames.com>

systems cannot adapt to the different requirements of various virtual environments. The rendering adapter enables a specific engine to connect to the generic S-MADE runtime. It maps agent goals and actions into the API functions. For example, ActiveWorld uses functions written in C programming language thus it requires calling the dynamic link library to achieve the storytelling, while Torque 3D game engine uses torque script which can be run directly in the virtual storytelling. Therefore, rendering adapters for the two rendering engines had to be developed. In order to export a current interactive storytelling to a new virtual environment, there is a need to write the according rendering adapter for the virtual environment.

In Figure 6.1, the designer refers to the developer who creates the interactive virtual storytelling and the user is the player/audience to enjoy the interactive storytelling.

6.2.3 Fuzzy Cognitive Goal Net Designer

Fuzzy cognitive goal net designer (based on goal net designer developed by Yu [77]), provides an intuitive graphical user interface for storytelling developers (especially novices), to model the complex structure of story plots or character behaviors easily.

The main interface of fuzzy cognitive goal net designer is shown in Figure 6.2. Fuzzy Cognitive Goal Net Designer contains the following manipulations:

- *Create Goal*
 - Create a composite goal with a red “S” circle
 - Create an atomic goal with a green “S” circle
- *Create Transition*
 - Create a transition with a “T” circle
- *Create Task*
 - Open the function list, and drag the relevant function into the goal or transition as task

CHAPTER 6. AGENT-BASED INTERACTIVE STORYTELLING ENGINE: S-MADE

Figure 6.2: Interface of Fuzzy Cognitive Goal Net Designer (Left Panel: Buttons to Add Goal, Transition and Tasks; Right Panel: Drawing Panel of Fuzzy Cognitive Goal Net)

- *Create Fuzzy Cognitive Map*

Launch the Fuzzy Cognitive Map modeler to create the Fuzzy Cognitive Map about relevant variables

- *Delete a FCGN component*

Delete a goal or a transition

- *Rename a FCGN component*

Double click the component, and the properties can be changed in the property dialog

- *Adjust the position of a FCGN component*

Click the component and drag to the new position

By using Fuzzy Cognitive Goal Net designer, a storytelling developer is able to design a complex goal net for the director agent or character agent easily with simple drag and drop operations. As shown in the figure, composite goals are represented with red circles, and primitive goals are represented with green circles. The transitions between the goals are shown as round rectangles. Directed arcs connect the goals and the transitions. Dash lines are used to link a composite goal with its sub-goals.

Any composite goal must be complete, so that there must be at least a route of goals and transitions which connects from “start” goal to “end” goal.

6.2.4 S-MADE Runtime for Single Agent

Inside the agent factory, each agent is run by the S-MADE runtime simultaneously in parallel. Once a dummy agent is loaded with a fuzzy cognitive goal net, it will start to run in order to achieve its goals. A solid and efficient agent running environment is important to the design and implementation of an agent system. In S-MADE, a hybrid agent architecture is employed, i.e., the agent is both goal-oriented and reactive. The S-MADE runtime includes a number of units to handle goal-oriented behaviors and reactive behaviors, which is illustrated in Figure 6.3. The running environment has the following key units which run in parallel:

Figure 6.3: Units of S-MADE Runtime to Handle Single Agent

- *Data Unit*

It is the unit to manage the physical connection and data retrieval between an agent and the goal net data source and other persistent data sources. It is responsible for loading the goal net from the Goal Net storage before the agent starts running. The data source can be either in the form of Goal Net description language, e.g., XML file, or a pre-defined Goal Net database.

- *Process Unit*

Process Unit has two major functions. First, it works as a convertor to understand the Goal Net description loaded by Data Unit and convert different goal net components into appropriate objects. Second, it serves as a runtime Goal Net storage when agent is running.

- *Knowledge Unit*

This unit interacts with a knowledge base and processes the knowledge level infor-

mation such as agent's learning. It provides knowledge such as ontology for making good reasoning.

- *Compute Unit*

This unit conducts the computational work for the agent, which includes the reasoning for goal selection and action selection.

- *Perception Unit*

This unit monitors and perceives the change of the agent's virtual worlds, which are registered as events in this unit, as well as user interactions in real-time.

- *Control Unit*

This unit is the brain of the agent, which coordinates the operations of other units. It processes Goal Net to achieve the final goal, coordinates other units, and manages the overall operations of the agent.

- *Action Unit*

This unit is responsible for task execution of the agents in the virtual environment, in the process of goal pursuing.

- *Reasoning Unit*

The unit is responsible for reasoning the context changes in the virtual environment in order to execute a reasonable behavior of the agent.

- *Communication Unit*

This unit handles the interactions with other agents inside the agent factory. It controls the message sending, receiving, semantic understanding and sends to control unit to process.

6.2.5 S-MADE Running Process

When an agent starts running, it will first load its fuzzy cognitive goal net from the database through the Data Unit. The Data Unit establishes a physical connection between the agent

CHAPTER 6. AGENT-BASED INTERACTIVE STORYTELLING ENGINE: S-MADE

and a database. It queries the database upon the agent's requests and returns the query results to the agent. The Data Unit does not care what kind of data it is retrieving from the database. Instead, it only knows how to transform the agent's request into a proper database query and send it to the database driver.

The Process Unit is responsible for reconstructing the goal net. As discussed earlier, the Process Unit retrieves different goal net components and links them up to recover the goal net. These components are mapped to different objects, the overall directory of goals, transitions and task lists are maintained by the Process Unit. When an agent starts processing the goal net, there are four units involved: Process Unit, Control Unit, Compute Unit and Action Unit. The process can be summarized as follows:

- Control Unit queries the Process Unit about the next goals (there are more than one next goal available when there is a choice transition).
- Process Unit informs Control Unit about all the possible next goals.
- Control Unit sends these goals to Compute Unit
- Compute Unit selects one from all the validated goals according to the goal selection algorithms, and sends the selected goal back to Control Unit.
- Based on the selected next goal, Control Unit asks Process Unit about the transition that can reach the selected goal.
- Process Unit sends Control Unit the corresponding transition.
- Control Unit sends the transition to Compute Unit for action selection.
- Compute Unit finds the best action to execute, and sends this action back to Control Unit.
- Control Unit sends the selected task to Action Unit for execution.

- Action Unit executes the given task by dynamically invoke the task, and sends acknowledgment to Control Unit once the task is done.
- Control Unit transits from its current goal, and asks Process Unit about the next goals.

When encountering a composite goal, Process Unit will directly go into the sub-goal-net and return the first goal of this sub-goal-net when Control Unit queries it. When it finishes the sub-goal-net, which means the composite goal is achieved, Process Unit will continue with the next goal of the composite goal. During the goal running process, Control Unit works as man-in-middle among other three units. The centralized coordination allows the developers to easily control an agent's running state. For example, the developers can simply suspend an agent by informing the Control Unit to pause, and wake up the agent by resuming the Control Unit. Once the end goal of the goal net is achieved, it is considered that the overall goal is achieved. The agent will inform the user and enter a dummy state. MADE is used mainly for offline applications, while S-MADE is used for real-time interactions, therefore, there is a need to have a more efficient goal selection algorithm or a simplified structure.

6.3 Evaluation

The evaluation of interactive storytelling system is a challenge. This is because such an evaluation is very subjective, and different users may evaluate the product differently based on their different preferences. Murray [52] proposes three categories for analysis of interactive story experiences:

- **Immersion:** Immersion is the feeling of being involved in the environment and engaged in the interactions with the environment.
- **Agency:** Agency is the feeling that empowers the player to take actions in the environment according to its intention.

- **Transformation:** Transformation means that the player is able to transform to different roles to experience the storytelling.

The comparisons with other approaches and the user interaction evaluation are illustrated in details.

6.3.1 Comparisons with Other Storytelling Methodologies

The comparisons of the hybrid methodology with plot-based interactive storytelling (PBIS) and character-based interactive storytelling (CBIS) is shown in Table 6.1. Compared

Factors	PBIS	CBIS	Hybrid
Story Generation Autonomy	Yes	No	Yes
Character Performing Autonomy	No	Yes	Yes
Context Awareness	High Level	Low Level	High & Low Level
Immersion	Low	High	High
Agency	At Authoring	At Presenting	Both
Transformation	No	Limited	Yes

Table 6.1: Comparisons with Plot-based and Character-based Interactive Storytelling

with other interactive storytelling systems, goal-oriented agents provide a higher level of autonomy and more human-like than behavior oriented agents.

Façade [48] proposes a hybrid approach which uses a “middleware” to connect the story authoring and story presentation together. However, there isn’t a unified mechanism to describe the activities of the director and characters. Thus the design of director and characters are separated, which degrades the storytelling efficiency. In our approach, the activities of director and characters are modeled with goals. Through the goal decomposing, the director assigns the scenes to character in real time.

In MINSTREL [68], “Schema” is used to describe the behaviors for director and characters. In our designed engine, a graphical interface with goal description language is used, which provides a more intuitive experience for designers to design interactive storytelling scenarios.

6.3.2 Comparisons with Other Storytelling Systems

There are some storytelling systems or agent systems in the market that can be considered for agent-based interactive storytelling.

Java Agent Development Environment (JADE) is a cross-platform multi-agent architecture. The developers can create an agent factory with multiple behavior-customized agents very easily. Compared to it, S-MADE has done more customization to create interactive storytelling besides the same easy agent creation. Moreover, it includes the intuitive story designer which helps the novice to build complex story scenario easily.

Storytelling Alice is also one famous story authoring tool which is easily used by students. However, it targets for students to develop simple stories with simple interactions to learning programming, which only deploys the story in its own virtual environment. By using S-MADE, the designers can design the interactive storytelling for different virtual environment after implementing different virtual environment adaptors. As the story development is separated from the story execution, the story is re-usable for other story development and for other virtual environment.

6.3.3 User Interaction Evaluation

The ultimate goal for interactive storytelling is to enhance the experience of the audience by enabling interaction. Therefore, there is a need to discuss interaction capabilities within the hybrid system. The role of the audience/user in the storytelling determines the interactions he/she can make. In the proposed hybrid system, the audience/user is able to act in different roles, and interact at different levels, as shown in Table 6.2.

Role of User	View	Level of Interactivity
Spectator/Observer	Third-Person View	Low-level (Indirect)
Director	Top View	High Level
Actor	First-Person View	Low-level (Direct)

Table 6.2: User Interaction Levels

As in conventional storytelling and some interactive storytelling systems, the audience is a spectator/observer with a third-person view. As an observer, there is no task or behavior assigned by the director. Therefore, he/she can only watch the performance of the virtual actors, and have limited interactions with the virtual environment. For example, he/she can change the position or availability of some objects required by the plot.

Moreover, according to the definition of the interactive storytelling, the audience is able to participate as a protagonist. The audience is able to interact with other virtual actors directly and contribute to the progress of the storytelling. The first-person view and the direct interaction increase the experience of immersion greatly.

Lastly, the system enables the audience to perform as a director or to interact with the director as interactive story authoring systems, which selects the storyline dynamically based on audience preference or context changes, and assigns the tasks to actors for performing. The story authoring allows the audience act at high level design over the story scenario, including choosing the storyline and the characters.

6.4 Summary

In this chapter, an agent-based interactive storytelling engine S-MADE is developed to facilitate the novice developers to generate interactive storytelling system from a text-based story. With the help of S-MADE, designers are able to design and create a workable interactive storytelling product easily.

In order to test our methodology, we illustrate two applications implemented with S-MADE in next chapter, which are implemented to suit science learning at secondary level. Combining the interactive storytelling in the virtual environment with science education promotes the learning interests of students, and enables the deeper learning.

CHAPTER 7

APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

7.1 Introduction

Storytelling as a way to deliver knowledge has been investigated by a lot of educators and researchers to promote learning interest of students. Virtual learning environments have been proven to be a new platform for students to conduct learning efficiently [22]. Agent-based learning environments have been developed as a research tool for investigating teaching and learning [4]. Compared to conventional classroom learning, interactive storytelling in a virtual learning environment combines the advantage of storytelling and virtual learning environment, which enables the students to explore the knowledge immersively.

In our story-driven learning, the learning contents are encapsulated in the storyline seamlessly. Related science concepts are linked together in the story. For example, the students may learn how to save a banana tree by investigating the plant transportation system; at receiving the reward “banana”, a virtual adventure is carried on at studying the human digestion system. As a new research tool to help teaching and learning, interactive storytelling in the virtual environment fosters the learning interests of students and motivates the deeper learning.

In this chapter, we illustrate two case studies which were developed to use the agent-based

interactive storytelling methodology to teach science subjects. The two case studies belong to two research projects respectively, which are 1) “Virtual Singapura”: the students return to 19th century Singapore to help investigating a mystery infectious disease; and 2) “Chronicles of Singapura”: the students save the dying banana trees by their adventures through the plant transportation system of them.

7.2 Virtual Singapura

7.2.1 Project Background

The purpose of the Virtual Singapura (VS) project is to conduct research into how agent-based multi-user virtual environments (A-MUVE) may be used to engage and motivate students at the lower secondary level in Singapore as they learn important scientific knowledge and skills, and is inspired by the River City project at Harvard University [20].

The story scenario of the Virtual Singapura MUVE is a historical context in 19th century Singapore. The synthetic characters in this environment are augmented with advanced agent technologies. Experimental and classroom research involving students’ learning with VS are being conducted at the secondary level. Assessments focus on whether students who use this multi-user virtual environment can construct deep understandings of scientific concepts and develop science inquiry skills, as well as the ability to transfer or apply their new knowledge and skills to new situations. This research also investigates the potential of this environment to motivate underachieving students. In addition, findings from this project will aid in determining if there is a gap between current testing of science knowledge and skills conducted in Singapore schools and the alternative, more authentic assessments of learning that will be employed in the project. It is hoped that this project will provide a foundation for future research into science learning with agent-based multi-user virtual worlds in a larger numbers of Singapore classrooms and schools as well as future international collaborative research projects.

The story scenario of *Virtual Singapura* is:

“...Nanyang town was a small town in 19th century’s Singapore, a small river crosses the town. People from different races lived together in harmony. Recently, a mystery illness spread all over the town. A lot of people were sick and even some of them died. Nobody knew what kind of illness it was, where it came from and how to stop it from spreading further. Even the most famous doctor Dr. Vajabali was not able to get any clue. A lot of rumors raised everywhere. There was a need to stop this situation as soon as possible. A group of scholars were sent to the town by the government to investigate the illness. Once they had arrived the town, they distributed the tasks and went to different places to investigate immediately. Mike and Jennifer met Dr. Vajabali and the nurse Miss Siti in the hospital, and visited the patients there. From there, they got some symptoms of the infectious illness. Most of them got high fever continuously in days time, some had cough, some got severe bleeding at cough. John went to the local laboratories, where air samples and water samples were collected to be examined. However, it required nearly half a month to process. Having got some clues, the scholars went to see Sir Andrew Clarks, who worked in Ministry of Health. By examining numerous books and collections together, some possibilities were identified. After the sample testing results were out, it was known that the water for wells in the town was polluted. They needed to go to the town as soon as possible to stop people from drinking the water...”

7.2.2 Story Implementation

The aim of the story is to teach secondary students about infectious diseases in normal life by exploring the virtual town and investigating the mystery illness. By talking to different non-player characters (NPC) or through conducting lab experiments in the town, the investigators need to find the symptoms of the mystery illness, study the differences among various diseases, and conclude with a thorough review of the mystery illness. The investigators’ results include name of the disease, symptoms, and precautions. In the story, the investigators can also apply their knowledge about illness in the virtual world, e.g., cleaning up dirt which might pollute the water, teaching the residences precaution methods.

The scriptwriter agent models the story scenario as shown in Figure 7.1. The investigator has many choices in the investigation. For example, he/she can go to either the hospital or the clinic to check the symptoms of the mystery illness and how widely the illness is spread. Depending on the availability of the officer at the health ministry, the investigator can choose to ask for differences between various illnesses from officer, or go to library to

Figure 7.1: Fuzzy Cognitive Goal Net for Illness Investigation Scenario by the Scriptwriter Agent

check from the books. Moreover, he/she can go to the town to verify his/her conclusion about the mystery illness, or do some further laboratory tests.

Figure 7.2: Storyline Generated by the Director Agent (Marked in the Dark Color)

The director agent selects a storyline dynamically based on the user interactions and current context. A selected storyline is as shown in Figure 7.2 with goals marked in the dark color, in which the investigator visited the hospital for illness symptoms, then went to meet the officer in Ministry of Health to query about the differences of the illnesses, and finally went to visit the town to confirm the his/her conclusion. Figure 7.3 shows a detailed fuzzy cognitive goal net of scene “visit hospital” in Figure 7.2. The scene involves three characters: a doctor, a nurse and the investigator. They perform the behaviors in parallel represented by the parallel transitions in the goal net.

Figure 7.3: “Visit Hospital” Scene

As shown from Figures 7.4 (a) to (g), the director agent dispatches the tasks of visiting different places to the three characters including the doctor, the nurse and the investigator. The player who acts as the investigator is able to interact in either the first-person view or third-person view.

The Fuzzy Cognitive Map used in the scenario is shown in Figure 7.5. The figure shows the relationships among several concepts. Among the concepts, “distance to town”, “distance to river” and “energetic” are context variables; the concept “cup available” can be modified by the user interaction; and the concepts “visit town” and “lab test” are two goals shown in Figure 7.3. The director agent needs to decide whether “visit town” is better than “lab test”.

Suppose the initial state vector from concept C_1 to C_6 is set as

$$(0.8 \ 0.2 \ 0.4 \ 1.0 \ 0.0 \ 0.0)$$

And the weight matrix (row and column are both from concept C_1 to C_6) derived from expert knowledge can be computed as

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0.5 & -0.5 \\ 0 & 0 & 0 & 0 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0.4 & -0.2 \\ 0 & 0 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

Figure 7.4: Screenshots within the Hospital Scene

Figure 7.5: Fuzzy Cognitive Map in the Hospital Scene

The output state vector from concept C_1 to C_6 after the FCM running is

$$(0.8 \quad 0.2 \quad 0.4 \quad 1.0 \quad -0.38 \quad 1.16)$$

It shows that, the value of C_6 is much higher than the value of C_5 . Thus, “lab test” is the final goal for the director agent at the goal selection based on the user interaction and context. In this example, the Fuzzy Cognitive Map does not have any loop. If there is any loop in causal relationships among the concepts in a fuzzy cognitive map, the state vector might converge in several steps, and the final decision is made based on the final result.

7.2.3 Results and Discussions

7.2.3.1 Pilot Study Results in Educational View

A pilot study was implemented over 10 class periods for secondary one students in Singapore. The aim was to check whether storytelling in the virtual environment was better at helping the students on science inquiry than the paper-based lessons.

Two top secondary schools (i.e. Catholic High Secondary School and Xin Min Secondary School) are chosen to evaluate the agent-based virtual learning environment. A total of 327 secondary two students (aged from 15 to 16) completed both the pretest and post-test, of which 105 were from the experimental group. To ensure equal sample sizes for analysis, 105 pupils were randomly selected from the comparison condition. A one factor

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

between (Group: Experimental or Comparison) x one factor within (Test performance: Pre and Post) repeated measures analysis of variance (ANOVA) was conducted on the parallel items of both the pretest and post-test. Results revealed a significant but weak Group effect ($F(1, 208) = 9.35, p < .01, \eta^2 = .04$). Further analysis using t-tests indicated that the students from Comparison group ($M = .78, SD = .11$) did slightly better on the parallel items from the post-test than those from the Experimental Group ($M = .72, SD = .13; t(208) = -3.48, p < .01, d = .43$). There were no significant group differences on the remaining multiple choice and open-ended questions on the post-test.

When interviews were conducted on 20 students and 2 teachers about the learning/playing experience, the response was that 80% of the students found the science inquiry in the Virtual Singapura environment to be very engaging and fosters different learning experience. Here are comments from two interviewed students:

- *“...so we don’t feel bored...because the teachers just like (gave) us all the notes...then we just copy down and all that, but in “Virtual Singapura”, we get to carry out experiments ourselves, so we understand better.” - Student 1, who is 14 years old, male, and in secondary two*
- *“...we all learn how to use our information we gathered to err... like conduct experiments, form a hypothesis, and err...find the outcome of the experiment...” - Student 2, who is 14 years old, male, and in secondary two*

By involving the students into the storytelling in the virtual environment, they learn through exploration, which can be a very good complement to the conventional paper-based science inquiry.

Compared to another science inquiry project in the virtual environment, Harvard’s “River City” lead by Dede [20], Virtual Singapura has made the non-player characters more intelligent and interactive to the players. As a result, 75 % of the players (15 out of 20) can enjoy better situated learning with an engaging experience in Virtual Singapura.

7.2.3.2 Results in Technical View

By considering each entity as an agent, Fuzzy Cognitive Goal Net is used as a universal modeling tool, which improves the modeling efficiency. Goals for agents can be easily reused or inherited. For example, the goal “greet people” of the doctor agent can also be used by other agents (e.g. Nurse agent). A new “greet people” for a specific purpose can be created by extending an already available “greet people” goal, which saves the developing time greatly. Because both the director and characters are modeled in real-time, it makes the storytelling more dynamic and enables the user interactions in different levels. Therefore, the user can change the storyline and the behaviors of a single character in real-time. By modeling a user agent, S-MADE can provide a more personalized story based on the user profile and user interactions.

S-MADE system provides a very easy way to author a very complex story and deploy it into the virtual environment through the goal net designer and S-MADE runtime. As it works independently from the rendering engine, porting the storytelling to different virtual environment is easily achieved by creating the virtual environment interfaces.

7.3 Chronicles of Singapura

7.3.1 Project Background

The project *Chronicles of Singapura* aims to create multi-agent augmented virtual worlds as future learning platforms. Virtual environment as a new class of serious games have gained tremendous worldwide popularity among young users in the recent years. Their potential for innovative and ground breaking research has been recognized by leading scientists [2]. Preliminary studies on the use of virtual worlds as learning environments to promote highly immersive experiential learning have shown encouraging results [19]. The potential of developing a new industry around this paradigm shift in education is very promising.

Our work aims to add a new dimension to the current virtual world based learning environments, namely using software agents and interactive storytelling to incorporate cutting edge

learning theories into a visually appealing immersive environment to enhance the learning process.

7.3.2 Agent-based Virtual Learning Environment (AVLE) with Interactive Storytelling

Virtual laboratories provide a new approach for students to learn knowledge, which might not be easy to carry out in the real-life experiments due to factors such as physical limitation, building cost etc. However, each student is unique, in terms of the learning curve of new knowledge, learning habit etc. A generic virtual learning environment or virtual laboratory might not suit the needs of all the students. Therefore, there is a need to find a way to customize the virtual learning experience to the learners.

7.3.2.1 Learning Structure

Providing a personalized learning experience is the key to promote the learning experience of a student at science acquisition. Agent-based Virtual Learning Environment (AVLE) augments the virtual learning environment with multiple intelligent agents, which provides personalized virtual learning for the students based on the agent reasoning over the students' preferences and real-time interactions. Figure 7.6 shows the learning structure of

Figure 7.6: Learning Structure in the Agent-based Virtual Learning Environment

an agent based virtual learning environment. In order to provide a personalized learning experience, the user preferences are first gathered off-line. After that, the students learn in

the agent mediated virtual learning environment, which contains two parts: virtual laboratory and role-playing learning. In the virtual laboratory, the students are able to conduct 2D or 3D simulation of science concepts, by acting as a “God”. By acting as a “Player”, the students are able to immerse through a role-playing learning. Interactive storytelling is used to motivate the students to have the role-playing learning. Agents percept the real-time interactions of the students, reason about them and act back to the virtual learning environment, i.e., to provide a unique learning experience.

7.3.2.2 Goal Oriented Agents

Agents are goal-oriented objects, which can be visible or invisible in the virtual environment. A capable agent is able to percept, reason and act in the virtual environment by defining the initial goals and cognitive variables. Fuzzy Cognitive Goal Net is used as the goal model for agents who act in an agent-based virtual learning environment; this will be explained in details in next section.

In the agent based virtual learning environment, the following agents help the students and analyze the learning process in real-time:

- **Instructor Agent** Each instructor agent is capable of providing instructions to the students. By monitoring the learning process of the students, the agent is able to tune the instructions in terms of difficulty level and instruction details.
- **Assessment Agent** An assessment agent assesses the learning progress of the students, in order to investigate the learning efficiency of the students. Then it will feedback to the instructor agent to adjust the virtual learning.
- **Inhabitant Agent** Inhabitant agents are the believable non-player characters used to deliver the learning contents in the virtual learning environments, and can take any form, for example, that of a human or even a tree.

The agents have the following properties:

- *Emotional*: The agents display emotions based on the interactions with students as a feedback to user interactions.
- *Interactive*: The agents are able to interact with the students in real-time.
- *Immersive*: The agents are able to “percept, reason and act” in real-time.

7.3.2.3 Role-playing Learning

In AVLE, role-playing learning is a main concept that allows the student to immerse into the virtual environment to learn the knowledge. Inhabitant agents are distributed in the learning environments to provide the related knowledge. Thus, the students need to compare, evaluate and induct the knowledge which were gathered in different places and times, helping them to achieve the deeper learning.

Storytelling is a common way to deliver the knowledge. Incorporating a story into the virtual learning environment will help motivate the students to acquire new knowledge.

7.3.3 Meet Teacher at School Before the Exploration

In order to prove our proposed design, we applied it into our interactive storytelling project, “Chronicles of Singapura”. The scene is that, “in front of a school, the teacher asks the students to explore the inside of a banana tree, to study the plant transportation system”. We design a very simple “School Scene” as the goal net shown in Figure 7.7. The scene involves a teacher and a number of students. At the start of storytelling, one DIRACT agent is created to execute the goal “School Scene”. Each complex scene (i.e. a composite goal) can be separated into goals of different characters. For example, the goal “Scene Initialization” is a composite goal that can be separated into atomic goals in hierarchy. We can either use one DIRACT agent to perform the whole storytelling, or involve more DIRACT agents acting together.

We create the DIRACT agents for both the teacher and the students. Through the goal dispatching algorithm, the goal nets for the teacher and students are illustrated in Figure 7.8 and Figure 7.9 respectively.

Figure 7.7: Goal Net for the School Scene

Figure 7.8: Goal Net for DIRACT Agent "Teacher"

Figure 7.9: Goal Net for DIRACT Agent "Student"

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

We run the DIRACT agents inside the Torque Game Engine virtual environment. Some screenshots are taken in Figure 7.10, corresponding to the goal pursuing by the teacher and students. From the trial run in Catholic High School, the test results show that the

(a): Scene Initialization (Teacher Wandering and Students Gathering) (b): Asking the Protagonist to Explore inside the Banana Tree

(c): Ask the Protagonist to Teach the Inner Structure of Tree

Figure 7.10: Snapshots for (a) Scene Initialization (Teacher Wandering and Students Gathering); (b) Asking the Protagonist to Explore inside the Banana Tree; and (c) Ask the Protagonist to Teach the Inner Structure of Tree

students preferred the learning experience with the intelligent virtual characters. Moreover, the teachers of the science class also enjoyed using the system to develop the goals and behaviors of the non-player characters to be used in the storytelling.

7.3.3.1 Comparison with Character-based Interactive Storytelling

We adopt a character-based interactive storytelling approach here, i.e., each character is modeled to execute its goals separately, each character can adapt to the context changes. However, each DIRACT agent in our approach is goal oriented rather than action oriented, and this exhibits a higher level of human like intelligence. Each agent does not act passively once their action plan is fixed, but is able to direct the story plot according to user interactions in real-time.

7.3.4 Adventure in the Plant Transportation System

7.3.4.1 Learning Content

The agent-based virtual learning environment is prepared for secondary level science learning in Singapore. The chapter is about plant transportation. Some learning concepts include:

- (1) Xylem and Phloem of Root, Stem and Leaf: the structures and functionalities of xylem and phloem inside the plant.
- (2) Osmosis and Diffusion: different movement methodologies of the water and mineral molecules.
- (3) Photosynthesis: how the energy and oxygen is generated inside the leaf with water.

7.3.4.2 Implementation

In order to motivate the students to learn the concepts in plant transportation, we generate a story scenario, namely “saving the dying banana tree”.

“The banana trees in Singapura town are sick. The farmer “Uncle Ben” asks the investigators to explore the inner structure of the tree through the plant transportation system, in order to save the banana tree.”

We have implemented our agent-based virtual learning environment with the support of Torque 3D Game Engine.

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

There are a set of agents involved in the virtual learning environment to facilitate the learning of a plant transportation system. Three typical agents that provide the personalized learning are illustrated here.

Lab supervisor “Miss Lee” is a tutor in the virtual laboratory, who decides the learning content based on the student level and preferences.

Figure 7.11: Fuzzy Cognitive Goal Net of Lab Supervisor Agent to Choose the Learning Content

The goal net used by the supervisor agent is shown as Figure 7.11. If the student is in the beginner level, she will lead the student to do the virtual experiment (e.g. diffusion or osmosis); otherwise, she will recommend the student to enter into the banana tree to watch the diffusion or osmosis process of water molecules at the root. Figure 7.12 shows that the lab supervisor “Miss Lee” greets the student with some introductions. Figure 7.13 illustrates a 2D diffusion simulation that the student can play with. Through this simulation, the students are able to learn how the ink molecules move during diffusion and the variables that might affect the movement speed of the molecules.

Besides simulations in the virtual laboratory, the students can also watch the diffusion or osmosis first hand in the root of the plant, something which is impossible to do in real life experiments. The director agent is a background agent that directs the whole role-playing learning. It provides hints and analyzes the students’ behaviors as the student is playing. The goal net used by the director agent is shown as Figure 7.14. It can schedule the student to talk to different non-player characters to find the sick banana tree to start the plant

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

Figure 7.12: Lab Supervisor “Miss Lee” Greets Students with an Introduction (‘Greet Student’ Goal in Figure 7.11)

Figure 7.13: Diffusion Experiment with 2D Simulation: Add Ink Drops to Observe the Movements of Molecules of Diffusion (‘Experiment’ Goal in Figure 7.11)

transportation journey. The “visit plant transportation” is a composite goal. When the

Figure 7.14: Fuzzy Cognitive Goal Net of Director Agent to Control the Role-playing of Students

director agent pursues this goal, it will load the sub-goals of the “visit plant transportation” goal, which is shown in Figure 7.15. Here are some screenshots of the students at play when

Figure 7.15: Sub-goal of Director Agent to Visit Plant Transportation

the goals of the director agent are executed. Figure 7.16 shows that the student is exploring the stem xylem by flying upward. Through this, the students are able to observe the inner structure of the stem xylem and the molecules that flow in it. Figure 7.17 shows the cut structure of the leaf, in which xylem is on top of the phloem. This is different from the cut structure at the root or at the stem. Figure 7.18 shows that the student pushes the water molecule towards a carbon-dioxide molecule to generate food in the leaf. Through this process, the students are able to learn about photosynthesis immersively.

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

Figure 7.16: Student is Flying Upwards inside of Stem Xylem ('Visit Stem' Goal in Figure 7.14)

Figure 7.17: Cross Structure of Leaf: Xylem on Top and Phloem at Bottom ('Visit Leaf' Goal in Figure 7.14)

Figure 7.18: The Player Pushes the Water Molecule Towards a Carbon-dioxide Molecule to Generate Food ('Generate Food' Goal in Figure 7.14)

An agent can be any goal oriented object, not only human like characters. The water molecule is an inhabitant agent in the learning adventure, which asks the students for help to take them into the leaf where the photosynthesis is carried out.

The goal net used by the water molecule agent is shown in Figure 7.19. Depending on the

Figure 7.19: Fuzzy Cognitive Goal Net of Water Molecule Agent to Go Through Root, Stem, Leaf and Generate the Food

learning scenario, the goal of the water molecule is composed of a linear series of goals.

7.3.5 Assessments

We implemented the agent-based virtual learning environment in Catholic High School, Singapore as part of the project “Chronicles of Singapura”. The students attending the assessment are secondary two students who are aged from 15 to 16. One group of 36 students (Group 1) used the agent-mediated virtual learning environment to learn while another group of 34 students (Group 2) used formal classroom learning as a comparison. Group 1 was allocated the same learning time as group 2, around 2 hours. After the learning session, both groups were given a multiple choice question test based on the plant transportation.

After the students played the scenario in the virtual world, an online survey was made to retrieve their feedback upon the playing experience. The results are subjective opinions of the students, ranging from “disagree strongly” to “agree strongly”. They are shown in Table 7.1.

From the survey results, it is seen that 78% (28 out of 36) of the students agree that the game is full of fun to play and helpful for science study to students. The game exhibits an imaginary world that promotes the students’ learning interests. The learning by playing is performed in an enjoyable way. The students could review their prior knowledge and strengthen their understandings during the game play. Moreover, 86% (31 out of 36) of the students admitted that it was a good way to study by finding answers based on prior knowledge.

7.3.5.1 Exam Results of the Two Teams

From the test result, the AVLE group obtained a mean score of 13.55 and the CL group obtained a mean score of 14.05. As shown in Figure 7.20, the learning result of the AVLE group is close to that of CL. Considering that the students needed to use around 1 hour to familiarize themselves with the virtual learning environment, the AVLE students were still able to get high score (only 0.5 less than CL students). Moreover, more students managed to get highest score (18 points) in AVLE group than those in CL group. Because the MCQ

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

	Disagree Strong- ly	Disagree	Disagree Some- what	No Opin- ion	Agree Some- what	Agree	Agree Strong- ly
The game has fun to play.		2.9%	5.7%	14.3%	40%	20%	17.1%
The game is useful for my science study.		2.9%	2.9%	11.4%	45.7%	25.7%	11.4%
The game can help me experience what is impossible in the real world.	2.9%		5.7%	22.9%	20%	25.7%	22.9%
I enjoyed learning in virtual singapura.			5.7%	17.1%	31.4%	28.6%	17.1%
I revised what I learnt in the game.	5.7%	2.9%	11.4%	20%	42.9%	8.6%	8.6%
The adventure diary is helpful for me to re-view my prior knowl-edge.		5.7%	11.4%	31.4%	25.7%	14.3%	11.4%
The review strength-ens my understanding and memory of the knowledge.		5.7%	5.7%	20%	11.4%	40%	17.1%
Finding answers based on my prior knowledge is a good way to study.		2.9%	2.9%	14.3%	11.4%	48.6%	20%

Table 7.1: Survey Results of Chronicles of Singapura in Catholic High School

Figure 7.20: Distribution of Scores in Agent-based Virtual Learning Environment (AVLE) and Classroom Learning (CL)

CHAPTER 7. APPLICATIONS: INTERACTIVE VIRTUAL STORYTELLING FOR EDUCATION

questions included some open questions that required reasoning, the students performed well as the AVLE encouraged exploration and thinking in the learning process, rather than memorizing the knowledge. Figure 7.21 shows the average score of each question in both

Figure 7.21: Average of Each Question in Agent-based Virtual Learning Environment (AVLE) and Classroom Learning (CL)

AVLE and CL. It was found that, students from the 2 groups performs better in terms of average score in different questions.

Some questions required the students to make the reasoning based on what they knew, e.g., question 21.

Suppose you killed the plant cell in the Figure of question 14A with poison (that does not destroy the cell membrane) and immediately placed the dead cell in a 25% saltwater solution.

1. Osmosis and diffusion would not occur.
2. Osmosis and diffusion would continue.
3. Only diffusion would continue.
4. Only osmosis would continue.

In this case, students in AVLE performed better (0.48 point higher on average). On the other hand, students in CL performed better in the questions which required memorizing. The students in AVLE might focus on the exploration process more than memorizing the terms, as agents help them to remember key terms.

7.3.5.2 Discussions

Through the test case, we found that the students were very interested and motivated in learning in the virtual environment. With the similar computer game experience, they adapted to the virtual learning environment very fast.

However, the test results were not as good as expected, which might be due to the following reasons:

- (1) The students had little training time to familiarize themselves with the virtual learning environment. They need more time to be comfortable with the learning method.
- (2) VLE is a good complement but not a replacement to classroom learning. Especially when we conducted the virtual learning with the teachers supervising, the students got panic easily.
- (3) As the students in the test group already had decent academic performance, they might not be very sensitive of the different learning methods.

7.4 Summary

In this chapter, we have adopted the agent-oriented methodology for interactive storytelling (AOMIS) to develop three applications for science learning in secondary schools. The results shows that the methodology helps the teachers and students construct the interactive storytelling in the virtual environment with meaningful storylines, moreover, the students benefit from the interactive storytelling and immersive learning.

In the next chapter, we will draw the conclusions of this thesis and make the future recommendation to extend the current research.

CHAPTER 8

CONCLUSIONS AND FUTURE RECOMMENDATIONS

In this chapter, we draw the conclusions of the thesis and recommend possible future work about interactive storytelling in the virtual environment.

8.1 Summary of Contributions

In this research, an agent-based methodology for interactive storytelling (AOMIS) has been developed to create interactive storytelling in a virtual environment. The methodology combines the advantages of plot-based and character-based interactive storytelling, so that both story plot coherence and character believability are achieved. The interactive storytelling is constructed based on a multi-agent system. Different from previous research efforts, story authoring and character behavior are modeled together based on goal pursuing of agents, which empowers the director and characters with goal-level intelligence and autonomy. The methodology involves two main models: 1) Fuzzy Cognitive Goal Net for plot model and character model; and 2) Evolutionary Fuzzy Cognitive Map for context model. Fuzzy Cognitive Goal Net (FCGN) is proposed as the goal model for intelligent director agent and characters agent for plot planning and act planning in the storytelling. The goal model provides the abstraction of agent intention/goal/behavior in goal level, with fuzzy cognitive abilities to handle real-time user interactions and dynamic context changes in the storytelling. The scriptwriter agent creates the nonlinear plot with possible

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

storylines, through which the director agent is able to make different choices according to user's dynamic interactions and context changes. The director agent dispatches the scenes (director's goal) to the virtual characters as their acts (actors' goal), and the virtual characters act based on the assigned goal and interact with each other dynamically. Evolutionary Fuzzy Cognitive Map is proposed to model the dynamic causal relationship among the user interactions, context changes, and agent goals etc., which enables the agents to make smart goal selection and action selection. Based on the proposed methodology, an interactive storytelling engine S-MADE has been developed to design different interactive storytelling scenarios easily. Compared with the plot-based approach and character-based approach, the proposed hybrid approach achieves the story authoring as well as character performing autonomy at the same time. Moreover, the audiences are able to make interactions at different levels of authoring and storytelling, which enhances the experience with a large degree of freedom.

The main contributions are summarized as follows:

(1) **Agent-oriented Methodology for Interactive Storytelling**

A new interactive storytelling methodology based on goal-oriented agents (AOMIS) has been proposed, that bridges the gap between plot-based and character-based interactive storytelling with goal-oriented agents. In this approach, both the story plot generation and character performing generation are realized in real time as goal pursuing of agents. Moreover, both story coherence to convey a value and believability of character performance can be achieved. By acting as or interacting with one of the agents (e.g. director, character), the audience is able to experience the storytelling with multiple degrees of freedom. Compared to action-oriented agents in other researches, goal-oriented agents exhibit higher believability with both goal automation and action automation.

(2) **Fuzzy Cognitive Goal Net**

Fuzzy Cognitive Goal Net (FCGN) model has been proposed as the story plot model

and the character model. In proposed interactive storytelling methodology, FCGN is used to plan plots for the director and to schedule behaviors for the characters. Fuzzy Cognitive Maps (FCMs) are used as the context model to model the dynamic causal relations among context variables, user interactions, and agent decisions. The promising fuzzy cognitive ability empowers the agents to reason in a complex, incomplete, and dynamically changing context.

(3) **Evolutionary Fuzzy Cognitive Map**

A believable virtual world has dynamic and evolving causal relationships among the elements in the world context, while Fuzzy Cognitive Map is not capable to model it. Evolutionary Fuzzy Cognitive Map has been proposed as the context model to model the probabilistic and fuzzy causal relationships among the context concepts, thus providing the basis of agent reasoning in order to provide personalized storytelling.

(4) **An interactive storytelling engine (S-MADE) to create interactive storytelling from scratch**

An interactive storytelling engine, namely Multi-Agent Development Environment for Interactive Storytelling (S-MADE), has been constructed to implement interactive storytelling agents with predefined story scenarios. It enables designers without programming skills to develop interactive storytelling scenario conveniently.

(5) **Apply agent-based interactive virtual storytelling methodology to promote teaching and learning**

Agent-oriented interactive virtual storytelling is created with our methodology for the science education to promote teaching and learning. By embedding the learning contents into the story, the students were able to obtain a deeper learning experience through their explorations and interactions with story characters. Agent-based interactive storytelling promotes the learners' interest as a good reinforcement to classroom learning.

8.2 Limitations

Fuzzy Cognitive Goal Net is used as the plot model and character model by modeling the director and the character's goals and actions. The hierarchical structure with inheritance of goals saves the developer from creating new goals etc. However, it exhibits the following limitations to be solved:

- The current replanning algorithm is still limited. One agent might fail to re-plan if the algorithm cannot find goals that connect between state to the end state.
- It is difficult to design the Fuzzy Cognitive Goal Net if the goal is very complex.
- There is a need to develop the goal net for the story plot manually by the developer. It is still difficult to generate the goal net automatically.

8.3 Future Recommendations

8.3.1 Refine Fuzzy Cognitive Goal Net Model

8.3.1.1 Develop Goal Replanning Mechanism

Fuzzy Cognitive Goal Net is used as the goal model for agents involved in the storytelling, e.g., director, virtual characters and so on. Currently, Goal Net is used to construct the story plot at the beginning during the designing phase rather than during runtime, which limits real-time goal planning in story authoring for the director and in story presentation for characters. To handle the runtime changes and user interactions better, the Goal Net for a specific agent can be very large and complex. Therefore, a re-planning algorithm is required to simplify the Goal Net. For Hierarchical Task Network a replanning algorithm is shown in [25] to allow dynamic task replanning in real-time. Similarly, in the goal modeling, there is a requirement to develop such replanning algorithm to reconfigure the goals of agents in real-time.

8.3.1.2 Refine Goal Splitting Algorithm

Scene dispatching is the most important step to link the director who does the story authoring to the characters that do the story presenting, in order to achieve story authoring and presenting together. In the goal net model, it is done through the goal splitting algorithm. However, the current algorithm is not well developed in the case of complex scene coherence and interaction synchronization.

8.3.1.3 Constrain Fuzzy Cognitive Goal Net Generation with Meta-rules

Fuzzy Cognitive Goal Net provides a very flexible way for designers to create an interactive storytelling. However, it may lead to improper storylines if there is a lack of rules or constraints over the moral/cognitive contents. There is a need to incorporate certain kinds of “domain/story independent goal-nets”, which are essentially concerned with general “meta-rules” for governing/constraining the generation of possible storylines. In this way, some endings will be more biased, depending on what and whom the interactive storyteller is communicating with, some endings would be more appropriate than the others, in terms of the moral/cognitive contents of a story and the corresponding readiness of users.

8.3.1.4 Expand Feature of User Awareness for Director/Characters

Creating a compelling experience for the audience in interactive storytelling is a challenging issue in interactive storytelling research. Different users have different preferences and interests. The interactive storytelling can be adjusted to be fast or slow, be simple or complex, in order to fit the users’ learning patterns. However, current interactive storytelling techniques only consider general differences among the users without a systematic analysis. The audience of interactive storytelling is different from users of normal applications (e.g desktop application, web application) as they are active players, who expect to experience and enjoy in real-time. As player-centered design gains focus in game and computer entertainment development, providing personalized scenarios is one of the most important aspects of player-centered design. Different users are comfortable with different complexities which are suited to their own requirements.

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

User awareness can be achieved by examining a user's profile from the knowledge base. There is also a need to learn the preferences and interests from the users' behaviors in the storytelling. For example, a user may have his/her preference over the length of narrative, or he/she might not want to wait for long sentences from the characters. Therefore, in real-time, it is a better choice to turn the narrative length to short. Reinforcement learning, e.g., Watkins' Q-Learning [72] can be used to learn the player's behaviors, with which scenarios can be provided with higher quality with rewards and punishments to enhance social experience for the player.

User awareness requires the researchers to study the users well. There is a need to analyze or learn the personalities of the users through their behaviors in the virtual environment, which helps the designers to provide them an interactive storytelling session with a more engaging experience.

Emotion modeling and personality modeling would enhance the compelling experience of users greatly. In the virtual environment, the behavior, personality and emotion are required to be modeled more consistently to make the agents become more believable in the storytelling process, so that the user can gain a more compelling experience with "reality".

8.3.1.5 Expand Feature of Context Awareness for Director/Characters

The director and virtual characters are intelligent not only at reasoning user interactions, but also at reasoning the context changes of the story and the virtual environment. Context awareness is another very important feature for a robust/realistic interactive storytelling system. It means that the situated environment affects the progress of the storyline. Context awareness is achieved through the reasoning or cognition ability of the agents. There is a need to provide more sophisticated mechanisms for reasoning. Context modeling is very important. Different models are available, e.g., Fuzzy Cognitive Maps [38], Ontology based context modeling [71], and so on. Besides the fuzziness, random events with probabilities are also needed to model a dynamic virtual environment.

8.3.2 Develop Objective Evaluation Metrics

As shown in Section 6.3, Murray [52] proposes three categories for analysis of interactive story experiences: immersion, agency and transformation. Other researchers also propose some more specific analysis methods [12].

However, these analysis methods are subjective to the audience, and so are not suitable for quantitative measurement of the storytelling. Until now, there is a lack of such objective methods to describe some properties for interactive storytelling, e.g., “interest”, complexity. Given this, the following parameters are possibly raised to justify some properties of the interactive storytelling, i.e., the complexity and reasonability:

- Number of characters
- Number of scenes
- Level of difficulties
- Number of choices per selection
- Relevance/reasonableness/completeness of choices
- Semantic relevance of the choices

However, audiences with different preferences are more favorable with different values of parameters. Therefore, it can be measured efficiently which plot is most suitable for specific

	Novice	Expert
Complexity	Low	High
Steps	Less	More
Choices	Less	More
Relevance	High	High

Table 8.1: Different User Preferences Cause Different Requirements

audience.

Moreover, the evaluation methods of school test need to be improved in order to evaluate the agent-based virtual learning environment (AVLE) in a long-run. In the previous school

test, AVLE is used to replace the classroom learning; while students have very limited time to be familiar with the virtual environment and learn inside. In the future, AVLE can be deployed as an informal learning method, thus the students can have more time to explore in the virtual learning environment, and to learn knowledge as well as problem solving skills immersive.

8.3.3 Text to Scene or Speech to Scene

Currently, the designers need to draw the goal nets for the agents within the story manually. Scene generation is very tedious for the developers, and the novices cannot do it easily in the virtual environment. We expect to generate a text-to-scene or speech-to-scene system to facilitate the scene development for both professional developers and novice users. The main advantages of text to scene and speech to scene are

- Improve the scene development efficiency. It provides a high-level and semantic interaction for developers to generate the story scenes.
- Simplify the scene development for novice developers. The scene can be generated with simple text input or speech input, with a certain format predefined.
- Fun interactivity for the students and teachers. It adopts an engaging human computer interface which is “what you see is what you get”.

The research work carried out in this thesis, and the list of the future work to be done have shown that the research and design of agent-based interactive storytelling in the virtual environment are both interesting and challenging. Moreover, it can be applied in various application domains of education and entertainment. The primary objective of this thesis, namely, to present an agent-mediated interactive storytelling from methodology, models to applications, has been met.

APPENDIX

List of Publications

Conference Papers

- [1] M. Ashoori, Chunyan Miao, and Yundong Cai. Socializing pedagogical agents for personalization in virtual learning environments. In *Proceedings of IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops*, pages 346–349, November 5–12, 2007.
- [2] Zhiqi Shen, Chunyan Miao, and Yundong Cai. Framework for agent augmented game development. In *The Second Annual Microsoft Academic Days Conference on Game Development*, Florida, USA, February 2007.
- [3] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, and Zhiqi Shen. Fuzzy cognitive goal net for interactive storytelling plot design. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, page 56, New York, NY, USA, June 2006. ACM.
- [5] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, and Zhiqi Shen. S-made: Interactive storytelling architecture through goal execution and decomposing. In *AAAI Fall Symposium on Intelligent Narrative Technologies*. AAAI Press, November 2007.
- [6] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, and Zhiqi Shen. Context modeling with evolutionary fuzzy cognitive map in interactive storytelling. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 2320–2325, JUN 2008.
- [7] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, and Zhiqi Shen. Modeling believable virtual characters with evolutionary fuzzy cognitive maps in interactive storytelling.

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

In *AAAI Spring Symposium on Intelligent Narrative Technologies II*. AAAI Press, November 2009.

- [8] H. L. Zhang, Z. Shen, X. Tao, C. Miao, B. Li, Ailiya, and Yundong Cai. Emotional agent in serious game (dino). In *Proceedings of The Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 1385–1386, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.
- [9] Yundong Cai, Zhiqi Shen, Chunyan Miao, and Ah-Hwee Tan. Direct: Direct and act in interactive storytelling. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 273–276, 2010.
- [10] Han Yu, Yundong Cai, Zhiqi Shen, Xuehong Tao, and Chunyan Miao. Agents as intelligent user interfaces for the net generation. In *Proceeding of the Fourteenth International Conference on Intelligent User Interfaces*, pages 429–430, New York, NY, USA, 2010. ACM.
- [11] Han Yu, Chunyan Miao, Xuehong Tao, Zhiqi Shen, Yundong Cai, Boyang Li, and Yuan Miao. Teachable agents in virtual learning environments: a case study. In Theo Bastiaens, Jon Dron, and Cindy Xin, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009*, pages 1088–1096, Vancouver, Canada, October 2009. AACE.

Journal Papers

- [1] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, Zhiqi Shen, and Boyang Li. Creating an immersive game world with evolutionary fuzzy cognitive maps. *IEEE Computer Graphics and Applications*, 30(2):58–70, 2008.
- [4] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, and Zhiqi Shen. *A Hybrid of Plot-Based and Character-Based Interactive Storytelling*, volume 4469/2007 of *Lecture Notes in Computer Science*, chapter Technologies for E-Learning and Digital Entertainment, pages 260–273. Springer Berlin / Heidelberg, 2007.
- [2] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, and Zhiqi Shen. Cognitive teachable agent in virtual learning environment with fuzzy cognitive map. *International Journal of Information Technology*, 16:2, 2010.
- [3] Cai, Y.; Miao, C.; Tan, A.-H.; Shen, Z. and Yu, H. Author and Play: A Case Study of Agent-based Interactive Storytelling in “Virtual Singapura” Project. *IEEE Transactions on Learning Technology*, under review, 2010

A.2 Multi-agent Development Environment for Story-telling (S-MADE) Code

```

#include "TestCompute.h"
#include "State.h"
#include "Transition.h"
#include "TaskListTable.h"
#include "DataUnit.h"

int TestCompute::counter = 0;

void TestCompute::SendState(State* pState)
{
    //check whether the agent is involved in any fuzzy cognitive
    //goal net running or not
    if(SampleAgent::involved==1)
        printf("\n\n\n@@@@@@@@@@@@@@@@ yundong, i am involved.\n");
    else
        printf("\n\n\n@@@@@@@@@@@@@@@@ yundong, i am not involved.\n");

    if (0 == pState)
        return;

    ProcessUnit* pProcessUnit = GetProcessUnit();

    if (0 == pProcessUnit)
        return;

    //check which state is chosen
    if(pState->GetStateID()==pState->firstStateID)
    {
        if(SampleAgent::involved==1)
            m_pTransitionSelected = pProcessUnit->GetTransition(pState->
                GetOutputTransitionIDList().at(1));
        else
            m_pTransitionSelected = pProcessUnit->GetTransition(pState->
                GetOutputTransitionIDList().at(0));
    }
}

```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```

    }
    else
        m_pTransitionSelected = pProcessUnit->GetTransition(pState->
            GetOutputTransitionIDList().at(0));

        printf("The transition id is %d\n", m_pTransitionSelected->
            GetTransitionID());

    m_bBestTransition= true;
}

void TestCompute::SendTransition(Transition* pTransition)
{
    if (0 == pTransition)
        return;

    ProcessUnit* pProcessUnit = GetProcessUnit();

    if (0 == pProcessUnit)
        return;

    DataUnit* pDataUnit = pProcessUnit->GetDataUnit();

    if (0 == pDataUnit)
        return;

    TaskListTable tableTaskList(pDataUnit);

    m_pTaskSelected = 0;
    m_bBestTask= false;

    if (tableTaskList.Query(pTransition->GetTaskListID()))
    {
        int iTaskID = tableTaskList.GetRow();

        if (-1 == iTaskID)
            return;
    }
}

```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```
        m_pTaskSelected = pProcessUnit->GetTask(iTaskID);
        m_bBestTask= true;
    }
}
```

A.3 Evolutionary Fuzzy Cognitive Map Code

```
// Author: Cai Yundong
// Description: Running Evolutionary Fuzzy Cognitive Map
// File: EFCM.cs

using System;
using System.Collections.Generic;
using System.Collections;
using System.Text;
using System.IO;

namespace EFCM
{
    public class EFCM
    {
        public const int BIVALENT = 0;
        public const int TRIVALENT = 1;
        public const int LOGISTIC = 2;

        public ArrayList conceptList;
        // the sparse matrix is recorded as a list also
        public ArrayList arcList;
        public String listV = "";

        public EFCM()
        {
            conceptList = new ArrayList();
            arcList = new ArrayList();
        }

        public void printConceptList()
        {
            foreach (Concept c in conceptList)
            {
                c.print();
            }
        }
    }
}
```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```
}

public void printArcList()
{
    foreach(Arc a in arcList)
    {
        a.print();
    }
}

public double activate(double value, int mode)
{
    double dReturn;

    switch (mode)
    {
        case BIVALENT:
            // Simple binary threshold

            if (value > 0.0)
                dReturn = 1.0;
            else
                dReturn = 0.0;
            break;

        case TRIVALENT:
            // Range is [-1, 0, 1]

            if (value <= -0.5)
                return -1.0;
            else if (value >= 0.5)
                return 1.0;
            else
                return 0;

        case LOGISTIC:
            // Logistic transfer function for c = 5
```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```
        dReturn = 1.0 / (1 + Math.Exp(-5.0 * value));
        break;

    default:
        dReturn = value;
        break;
    }
    return dReturn;
}

//copy the arraylist of concepts
public ArrayList cloneArrayList(ArrayList conceptList)
{
    ArrayList tempList = new ArrayList();

    foreach (Concept c in conceptList)
    {
        Concept t=new Concept();
        t.id = c.id;
        t.x = c.x;
        t.y = c.y;
        t.value = c.value;
        t.deltaValue = c.deltaValue;
        t.time = c.time;
        tempList.Add(t);
    }
    return tempList;
}

//sort the concept list based on the index
/*
public void sortConceptList(ArrayList cList)
{
    for(int i=0; i<cList.Count; i++)
    {
        for(int j=i; j<cList.Count; j++)
        {
```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```

        Concept t1 = (Concept)cList[i];
        Concept t2 = (Concept)cList[j];

        if (t1.id < t2.id)
        {
            Concept temp = t2;

            t2 = t1;
            t1 = temp;
        }
    }
}*/

//UPDATE the values at this time, with different mode
public void update(int timeStamp, int mode)
{
    int TheSeed = (int)DateTime.Now.Ticks;
    Random aRand = new Random(TheSeed);
    double result = 0.0;
    ArrayList oldValue;
    double randvalue = 0.0;

    oldValue = cloneArrayList(conceptList);
    //oldValue = (ArrayList)conceptList.Clone();
    Console.WriteLine("Time: \t" + timeStamp);

    for (int i = 0; i < conceptList.Count; i++)
    {
        //check which concept need to be updated
        result = 0.0;

        Concept t = (Concept)oldValue[i];
        Concept n = (Concept)conceptList[i];
        //if (timeStamp % t.time == 0)
        {
            for (int j = 0; j < conceptList.Count; j++)

```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```

        {
            randvalue = aRand.NextDouble();

            Arc tempArc = getArc(j, i);
            Concept fr = (Concept)oldValue[j];
            if (tempArc != null)
                result += fr.deltaValue * tempArc.value;
        }

        Console.WriteLine("Change of " + n.name + " is " + result);

        n.deltaValue = result;
        //n.deltaValue = activate(result, TRIVALENT);

        n.value = activate(t.value + t.deltaValue, mode);
        //n.deltaValue = n.value - t.value;
        //n.value = n.value + n.deltaValue;
    }

    listV += "\t" + String.Format("{0:F2}", n.value);
    result = 0.0;
}
listV += "\r\n";
printConceptList();
}

public void update_old(int timeStamp, int mode)
{
    int TheSeed = (int)DateTime.Now.Ticks;
    Random aRand = new Random(TheSeed);
    double result = 0.0;
    ArrayList oldValue;
    double randvalue = 0.0;

    oldValue=(ArrayList)conceptList.Clone();
    Console.WriteLine("Time: \t" + timeStamp);

```


CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```

    for( int i=0; i<conceptList.Count; i++)
    {
        //check which concept need to be updated

        Concept t=(Concept)oldValue[i];
        Concept n = (Concept)conceptList[i];
        if(timestamp%t.time==0)
        {
            for( int j=0; j<conceptList.Count; j++)
            {
                randvalue=aRand.NextDouble();

                Arc tempArc=getArc(i, j);
                if(tempArc!=null && randvalue<tempArc.probability)
                    result+=t.value*tempArc.value;
            }

            n.value=activate(result,mode);
        }

        listV += "\t" + String.Format("{0:F2}",n.value);
        result=0.0;
    }

    listV += "\r\n";
    printConceptList();
}

//run the EFCM
public void run(int time, int mode, System.Windows.Forms.TextBox
output, System.Windows.Forms.ToolStripProgressBar progressbar)
{
    //add title
    listV = "Cycle\t";
    for (int i = 0; i < conceptList.Count; i++)
    {
        Concept t = (Concept)conceptList[i];

```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```
        listV += "\t" + t.name;
    }

    listV += "\r\n";

    output.Text = "";
    for(int i=1; i<=time; i++)
    {
        listV += i + "\t" ;
        update(i, mode);

        //update the progress bar
        progressbar.PerformStep();
    }

    //update the text
    output.Text = listV;
}

public Arc getArc(int i, int j)
{
    foreach(Arc a in arcList)
    {
        if (a.fromConcept == i && a.toConcept == j)
            return a;
    }

    return null;
}

public void loadFromFile(string filename)
{
}

public void writeToFile(string filename)
{
}
```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```
    }  
}  
  
// Author: Cai Yundong  
// Description: Running Evolutionary Fuzzy Cognitive Map  
// File: entity.cs  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace EFCM  
{  
    public class Entity  
    {  
        public int id;  
        public double value;  
        public double probability;  
  
        public Entity(int id)  
        {  
            this.id = id;  
            this.value = 0.0;  
            this.probability = 1.0;  
        }  
  
        public Entity()  
        {  
            this.id = 0;  
            this.value = 0.0;  
            this.probability = 1.0;  
        }  
  
        public Entity(int id, double value, double probability)  
        {  
            this.id = id;  
            this.value = value;  
            this.probability = probability;  
        }  
    }  
}
```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```
    }  
  }  
}  
  
// Author: Cai Yundong  
// Description: Running Evolutionary Fuzzy Cognitive Map  
// File: concept.cs  
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace EFCM  
{  
    public class Concept:Entity  
    {  
        public int time;  
  
        public int x;  
        public int y;  
        public string name;  
        public string description;  
        public double deltaValue;  
  
        public Concept():base()  
        {  
            time = 1;  
  
            x = 0;  
            y = 0;  
            name = "null";  
            description = "null";  
            deltaValue = 0;  
        }  
  
        public void print()  
        {
```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

```
        Console.WriteLine("Concept "+id+"\t"+value);
    }
}

// Author: Cai Yundong
// Description: Running Evolutionary Fuzzy Cognitive Map
// File: arc.cs
using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;

namespace EFCM
{
    public class Arc: Entity
    {
        public int fromConcept;
        public int toConcept;
        public Point fromPoint;
        public Point endPoint;

        public Arc(int id):base(id)
        {
            fromConcept = -1;
            toConcept = -1;
        }

        public void print()
        {
            Console.WriteLine("Arc " + id + ":");
            Console.WriteLine("\t" + value);
            Console.WriteLine("\t" + probability);
            Console.WriteLine("\t" + fromConcept);
            Console.WriteLine("\t" + toConcept);
        }
    }
}
```

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

A.4 Questionnaire for School Testing of Virtual Singapura

SECONDARY TWO

DIFFUSION, OSMOSIS AND TRANSPORT IN PLANTS

Section A: Multiple Choice Questions

Question 1 of 15

The movement of particles from the region of their higher concentration to the region of their lower concentration is called as _____.

1. diffusion
2. ascent of sap
3. osmosis
4. active transport

Correct Answer Number:

Question 2 of 15

Osmosis is a special case of :

1. cell membranes
2. diffusion
3. function
4. active transport

Correct Answer Number:

Question 3 of 15

Normally, in the process of osmosis, the net flow of water molecules into or out of the cell depends upon differences in the

1. concentration of water molecules inside and outside the cell
2. concentration of enzymes on either side of the cell membrane
3. rate of molecular motion on either side of the cell membrane
4. rate of movement of insoluble molecules inside the cell

Correct Answer Number:

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

Question 4 of 15

Minerals are absorbed into the cells by _____ .

1. osmosis
2. diffusion
3. active transport
4. 1,2 and 3

Correct Answer Number:

Question 5 of 15

Sodium ions are “pumped” from a region of lower concentration to a region of higher concentration in the nerve cells of humans. This process is an example of

1. diffusion
2. passive transport
3. osmosis
4. active transport

Correct Answer Number:

Question 6 of 15

The _____ transports water and mineral salts from the roots to the rest of the plant.

1. xylem
2. phloem
3. root hairs
4. vascular bundles

Correct Answer Number:

Question 7 of 15

The food is transported in the phloem in the form of _____ .

1. glucose
2. sucrose
3. amino acids
4. fats

Correct Answer Number:

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

Question 8 of 15

The ascent of sap in plants takes place due to _____ .

1. root pressure
2. transpiration pull
3. both 1 & 2
4. osmosis

Correct Answer Number:

Question 9 of 15

A. Suppose there is a large beaker full of clear water and a drop of blue dye is added to the beaker of water. Eventually the water will turn a light blue color. The process responsible for blue dye becoming evenly distributed throughout the water is:

1. osmosis
2. diffusion
3. a reaction between water and dye

Correct Answer Number:

B. The reason for my answer is because:

1. The lack of a membrane means that osmosis and diffusion can occur.
2. There is movement of particles between regions of different concentrations
3. The dye separates into small particles and mixes with water.
4. The water moves from one region to another.

Correct Answer Number:

Question 10 of 15

A. During the process of diffusion, the particles will generally move from:

1. high to low concentrations
2. low to high concentrations

Correct Answer Number:

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

B. The reason for my answer is because:

1. There are too many particles crowded into one area; therefore, they move to an area with more room.
2. Particles in areas of greater concentration are more likely to bounce toward other areas.
3. The particles tend to move until the two areas are isotonic, and then the particles stop moving.
4. There is a greater chance of the particles repelling each other.

Correct Answer Number:

Question 11 of 15

A. A glucose solution can be made more concentrated by:

1. adding more water
2. adding more glucose

Correct Answer Number:

B. The reason for my answer is because:

1. The more water there is, the more glucose it will take to saturate the solution.
2. Concentration means the dissolving of something.
3. It increases the number of dissolved particles.
4. For a solution to be more concentrated, one must add more liquid.

Correct Answer Number:

Question 12 of 15

A. If a small amount of sugar is added to a container of water and allowed to set for one week without stirring, the sugar molecules will:

1. be more concentrated on the bottom of the container
2. be evenly distributed throughout the container

Correct Answer Number:

B. The reason for my answer is because:

1. There is movement of particles from a high to low concentration.
2. The sugar is heavier than water and will sink.
3. Sugar dissolves poorly or not at all in water.
4. There will be more time for the sugar to settle to the bottom.

Correct Answer Number:

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

Question 13 of 15

- A. Two columns of water are separated by a membrane through which only water can pass. Side 1 contains dye and water; side 2 contains pure water. After 2 hours, the water level in side 1 will be:

1. higher
2. lower
3. the same height

Correct Answer Number:

- B. The reason for my answer is because:

1. Water will move from the hypertonic to hypotonic solution.
2. The concentration of water molecules is less on side 1.
3. Water will become isotonic.
4. Water moves from low to high concentrations.

Correct Answer Number:

Question 14 of 15

- A. The Figure below is a picture of a plant cell that lives in freshwater. If this cell were placed in a beaker of 25% saltwater solution, the central vacuole would:

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

1. increase in size
2. decrease in size
3. remain the same size

Correct Answer Number:

B. The reason for my answer is because:

1. Salt absorbs the water from the central vacuole.
2. Water will move from the vacuole to the saltwater solution.
3. The salt will enter the vacuole.
4. Salt solution outside the cell cannot affect the vacuole inside the cell.

Correct Answer Number:

Question 15 of 15

A. Suppose you killed the plant cell in the Figure of question 14A with poison (that does not destroy the cell membrane) and immediately placed the dead cell in a 25% saltwater solution.

1. Osmosis and diffusion would not occur.
2. Osmosis and diffusion would continue.
3. Only diffusion would continue.
4. Only osmosis would continue.

Correct Answer Number:

B. The reason for my answer is because:

1. The cell would stop functioning.
2. The cell does not have to be alive.
3. Osmosis is not random, whereas diffusion is a random process.
4. Osmosis and diffusion require cell energy.

Correct Answer Number:

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

Section B: Free Response Questions

1 Two pieces of potato strips, of the same length and thickness, are left in two petri dishes for two hours. One dish contains salt solution while the other contains water only.

a Draw the potato strips to show the results of the experiment.

b Explain your answer.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

CHAPTER 8. CONCLUSIONS AND FUTURE RECOMMENDATIONS

2. Four green plants were used to investigate the rate of water loss and the rate of water uptake by green plants when some of the plants' leaves are coated with grease. Each plant was weighed together with its test tube at the start of the experiment and again 24 hours later. The results are shown in the table below:

	Mass of Plant and Test tube (g)			
	Plant 1	Plant 2	Plant 3	Plant 4
Start of Experiment	120	108	116	119
24 hours later	101	106	98	118

Which of the plants have all their leaves coated with grease? Explain your answer.

.....

.....

.....

.....

BIBLIOGRAPHY

- [1] Aristotle. *Poetics*. The Loeb Classical Library, 300BC.
- [2] William Sims Bainbridge. The scientific research potential of virtual worlds. *Science*, 317(5837):472–476, Jul 2007.
- [3] Mieke Bal. *Narratology: introduction to the theory of narrative*. University of Toronto Press, January 1997.
- [4] AL Baylor. Agent-based learning environments as a research tool for investigating teaching and learning. 26(3):227–248, 2002.
- [5] Michael E. Bratman, David Israel, and Martha Pollack. Plans and resource-bounded practical reasoning. In Robert Cummins and John L. Pollock, editors, *Philosophy and AI: Essays at the Interface*, pages 1–22. The MIT Press, Cambridge, Massachusetts, 1991.
- [6] Selmer Bringsjord and David A. Ferrucci. *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, a Storytelling Machine*. Lawrence Erlbaum Associates, September 1999.
- [7] Rodney A. Brooks. Intelligence without reason. In John Myopoulos and Ray Reiter, editors, *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 569–595, Sydney, Australia, 1991. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.
- [8] Rodney A. Brooks. Intelligence without representation. 47(1-3):139–159, 1991.
- [9] Birgit Burmeister and Kurt Sundermeyer. Cooperative problem-solving guided by intentions and perception. *ACM SIGOIS Bulletin*, 13(3):10, 1992.

BIBLIOGRAPHY

- [10] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, and Zhiqi Shen. Modeling believable virtual characters with evolutionary fuzzy cognitive maps in interactive storytelling. In *AAAI Spring Symposium on Intelligent Narrative Technologies II*. AAAI Press, November 2009.
- [11] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, Zhiqi Shen, and Boyang Li. Creating an immersive game world with evolutionary fuzzy cognitive maps. *IEEE Computer Graphics and Applications*, 30(2):58–70, 2008.
- [12] Cungeng Cao, Xiaoli Yue, and Yuefei Xu. Pnai: A platform for narrative and animation intelligence. *Information Technology Letter*, 4(5):1–4, 2006.
- [13] J. Carvalho and J. Tom. Rule based fuzzy cognitive maps – qualitative systems dynamics. In *Proceedings of the Nineteenth International Conference of the North American Fuzzy Information Processing Society*, pages 407–411, Atlanta, US, 2000.
- [14] M. Cavazza, F. Charles, and S.J. Mead. Character-based interactive storytelling. *IEEE Intelligent Systems*, 17(4):17–24, 2002.
- [15] Marc Cavazza, Fred Charles, and Steven J. Mead. Interacting with virtual characters in interactive storytelling. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 318 – 325, Bologna, Italy, 2002.
- [16] D. Chapman and P. E. Agre. Abstract reasoning as emergent from concrete activity. In M. P. Georgeff and A. L. Lansky, editors, *Reasoning about Actions and Plans*, pages 411–424. Kaufmann, Los Altos, CA, 1987.
- [17] Chris Crawford. *Chris Crawford on Interactive Storytelling*. New Riders Games, October 2004.
- [18] Richard Dawkins. *The Blind Watchmaker*. W. W. Norton & Company, Inc., New York, NY, USA, September 1986.
- [19] Chris Dede. Immersive interfaces for engagement and learning. *Science*, 323(5910):66–69, Jan 2009.
- [20] Chris Dede, Brian Nelson, Diane Jass Ketelhut, Jody Clarke, and Cassie Bowman. Design-based research strategies for studying situated learning in a multi-user virtual environment. In *Proceedings of the Sixth International Conference on Learning Sciences*, pages 158–165. International Society of the Learning Sciences, 2004.
- [21] J. A. Dickerson and B. Kosko. Virtual worlds as fuzzy cognitive maps. In Bart Kosko, editor, *Fuzzy Engineering*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.

BIBLIOGRAPHY

- [22] P. Dillenbourg, D.K. Schneider, and P Synteta. Virtual learning environments. In *Proceedings of the Third Hellenic Conference Information & Communication Technologies in Education*, pages 3–18, 2002.
- [23] Magy Seif El-Nasr, John Yen, and Thomas R. Ioerger. Flame—fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, 3(3):219–257, 2000.
- [24] Clark Davidson Elliott. *The affective reasoner: a process model of emotions in a multi-agent system*. PhD thesis, Northwestern University, USA, Evanston, IL, USA, 1992.
- [25] Kutluhan Erol, James Hendler, and Dana S. Nau. Htn planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume 2, pages 1123–1128, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.
- [26] J. Ferber. Simulating with reactive agents. In E. Hillebrand and Eds J. Stender, editors, *Many Agent Simulation and Artificial Life*, page 8C28, Amsterdam, 1994. IOS Press.
- [27] I. A. Ferguson. *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. PhD thesis, University of Cambridge, Cambridge, UK, 1992.
- [28] Elizabeth Figa and Paul Tarau. The vista architecture: experiencing stories through virtual storytelling agents. *ACM SIGGROUP Bulletin*, 23(2):27–28, 2002.
- [29] Karen A. Frenkel. The next generation of interactive technologies. *Communications of the ACM*, 32(7):872–881, 1989.
- [30] M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 198–204. American Association of Artificial Intelligence, 1987.
- [31] Jonathan Gratch and Stacy Marsella. Tears and fears: modeling emotions and emotional behaviors in synthetic agents. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 278–285, New York, NY, USA, 2001. ACM.
- [32] Marde Greeff and Vali Lalioti. Interactive cultural experiences using virtual identities. In *International Cultural Heritage Informatics Meetings*, pages 455–465, Milano, Italy, 2001.
- [33] N. Hawes. Anytime planning for agent behaviour. In *Proceedings of the Twelfth Workshop of the UK Planning and Scheduling Special Interest Group*, pages 157–166, 2001.

BIBLIOGRAPHY

- [34] D.K.J. Heylen, A. Nijholt, and M. Poel. Embodied agents in virtual environments: The aveiro project. In K. Leiviska, editor, *Proceedings of European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems*, page 7, Tenerife, Spain, December 2001. NoE Eunite.
- [35] P. Hoffmann and M. Herczeg. Expanding the story line. *Museums and the Web*, pages 24–26, 2004.
- [36] N. R. Jennings. Specification and implementation of a belief-desire-joint-intention architecture for collaborative problem solving. *International Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318, 1993.
- [37] Caitlin Kelleher, Randy Pausch, and Sara Kiesler. Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1455–1464, New York, NY, USA, 2007. ACM.
- [38] Bart Kosko. Fuzzy cognitive maps. *International Journal of Man Machine Studies*, 24(1):66–75, 1986.
- [39] Bart Kosko. Fuzziness vs. probability. *International Journal of General Systems*, 17(2):211–240, 1990.
- [40] M. Lebowitz. Creating characters in a story-telling universe. *Poetics*, 13:171–194, 1984.
- [41] M. Lebowitz. Story-telling as planning and learning. *Poetics*, 14(6):483–502, 1985.
- [42] Dongtao Li. A goal-net based agent development environment and its applications. Master’s thesis, Nanyang Technological University, Singapore, 2006.
- [43] Pattie Maes. The agent network architecture (ana). *ACM SIGART Bulletin*, 2(4):115–120, 1991.
- [44] B Magerko and J.E. Laird. Building an interactive drama architecture. In *Proceedings of the First International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pages 226–237, Darmstadt, Germany, 2003.
- [45] Jacques Tisseau Marc Parenthoen, Patrick Reignier. Put fuzzy cognitive maps to work in virtual worlds. In *Proceedings of The Tenth IEEE International Conference on Fuzzy Systems*, volume 1, page 38, Australia, 2001.

BIBLIOGRAPHY

- [46] M. Mateas. An oz-centric review of interactive drama and believable agents. In *AI Today: Recent Trends and Developments, Lecture Notes in Artificial Intelligence*, volume 1600, pages 297–328, Berlin, 1999. Springer-Verlag.
- [47] Michael Mateas and Andrew Stern. Towards integrating plot and character for interactive drama. In *Working notes of the Social Intelligent Agents: The Human in the Loop Symposium. AAAI Fall Symposium Series*. AAAI Press, November 2000.
- [48] Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference*, 2003.
- [49] N. Mateou, A. Andreou, and C. Stylianou. Evolutionary multilayered fuzzy cognitive maps: A hybrid system design to handle large-scale, complex, real-world problems. *Information and Communication Technologies*, 1:1663–1668, 2006.
- [50] James Richard Meehan. *The metanovel: writing stories by computer*. PhD thesis, Yale University, USA, 1976.
- [51] Yuan Miao, Zhiqiang Liu, and Chunyan Miao. Dynamical cognitive network, an extension of fuzzy cognitive maps. *IEEE Transactions on Fuzzy Systems*, 9(5):760–770, 2001.
- [52] J. Murray. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. The Free Press, New York, NY, USA, 1997.
- [53] J. Murray. Hamlet on the holodeck. *Cambridge, MA: MIT Press*, 1998.
- [54] James Odell. Objects and agents compared. *Journal of Object Technology*, 1(1):41–53, 2002.
- [55] Vladimir A. Propp. *Morphology of the Folktale*. Publications of the American Folklore Society. University of Texas Press, 1928.
- [56] A. S. Rao and M. P. Georgeff. Bdi-agents: from theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, 1995. AAAI Press.
- [57] Mark Owen Riedl. *Narrative Generation: Balancing Plot and Character*. PhD thesis, North Carolina State University, 2004.
- [58] Ted Selker. Coach: A teaching agent that learns. *Communications of ACM*, 37(7):92–99, 1994.

BIBLIOGRAPHY

- [59] Zhigqi Shen, R. Gay, Yuan Maio, and Chunyan Miao. Goal autonomous agent architecture. In *Proceedings of the Twenty-Eighth Annual International Computer Software and Applications Conference*, volume 2, pages 45–46, 2004.
- [60] Zhiqi Shen, Chunyan Miao, Yuan Miao, Xuehong Tao, and R. Gay. A goal-oriented approach to goal selection and action selection. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 114–121. American Association of Artificial Intelligence, 2006.
- [61] Yoav Shoham. Agent oriented programming. In *Logic at Work*, pages 123–129, 1992.
- [62] David Canfield Smith, Allen Cypher, and James C. Spohrer. Kidsim: Programming agents without a programming language. *Communications of ACM*, 37(7):54–67, 1994.
- [63] Heng-Jie Song, Zhi-Qi Shen, Chun-Yan Miao, Zhi-Qiang Liu, and Yuan Miao. Probabilistic fuzzy cognitive map. In *Proceedings of IEEE International Conference on Fuzzy Systems*, pages 1221 – 1228, 2006.
- [64] Story.(n.d.). Kernerman english multilingual dictionary. retrieved july 23, 2007, from dictionary.com website: <http://dictionary.reference.com/browse/story>.
- [65] Storytelling. <http://en.wikipedia.org/wiki/storytelling>.
- [66] Storytelling.(n.d.). Webster’s revised unabridged dictionary. retrieved july 23, 2007, from dictionary.com website: <http://dictionary.reference.com/browse/storytelling>.
- [67] M. Theune, S. Faas, D.K.J. Heylen, and A. Nijholt. The virtual storyteller: Story creation by intelligent agents. In S. Göbel, N. Braun, U. Spierling, J. Dechau, and H. Diener, editors, *Technologies for Interactive Digital Storytelling and Entertainment*, pages 204–215, Darmstadt, Germany, March 2003. Fraunhofer IRB Verlag.
- [68] S. R. Turner. *MINSTREL: A Computer Model of Creativity and Storytelling*. PhD thesis, University of California LA, 1993.
- [69] Juan D. Velásquez and Pattie Maes. Cathexis: a computational model of emotions. In *Proceedings of the First International Conference on Autonomous Agents*, pages 518–519, New York, NY, USA, 1997. ACM.
- [70] Steven Vere and Timothy Bickmore. A basic agent. *Computational Intelligence*, 6(1):41–60, 1990.

BIBLIOGRAPHY

- [71] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung. Ontology based context modeling and reasoning using owl. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 18–22, 2004.
- [72] Christopher J. C. H. Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.
- [73] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.
- [74] Rafael Pérez y Pérez and Mike Sharples. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, 13(2):119–139, 2001.
- [75] R. Michael Young, Mark O. Riedl, Mark Branly, Arnav Jhala, R. J. Martin, and C. J. Saretto. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1(1):51–70, 2004.
- [76] R.M. Young. An overview of the mimesis architecture: Integrating narrative control into a gaming environment. In *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*. AAAI Press, 2001.
- [77] Han Yu, Zhiqi Shen, and Chunyan Miao. A goal-oriented development tool to automate the incorporation of intelligent agents into interactive digital media applications. *Computers in Entertainment*, 6(2):1–15, 2008.
- [78] Shen Zhiqi. *Goal-Oriented Modeling For Intelligent Agents and Their Applications*. PhD thesis, Nanyang Technological University, 2003.
- [79] Michael Zyda. From visual simulation to virtual reality to games. *Computer*, 38(9):25–32, 2005.