INTELLIGENT PEDAGOGICAL AGENTS
WITH
MULTIPARTY INTERACTION SUPPORT

Liu Yi

(B.Comp.(Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2005

ACKNOWLEDGEMENT

First of all, I sincerely thank and appreciate my supervisor A/P Chee Yam San for his guidance, encouragement and patience over the years. I was enlightened by him not only about the practical approaches of the research, but also the essence on lifelong self improvement. The more he has taught me, the more I feel I have to learn.

My gratitude also goes to Mr. Hooi Chit Meng, the predecessor of the C-VISions research project. Without him, I cannot imagine how I can survive at the beginning of the research.

I am also grateful to the members in LELS Lab: Yuan Xiang, Chao Chun, Jonathan, Lai Kuan, Lei Lei etc. It was an enjoyable and memorable experience studying in this lab.

Finally, it is time to thank my parents and all the friends. Their constant support and considerations have made my heart warm and enlightened throughout the whole period.

Liu Yi

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SUMMARY

Virtual learning worlds with embodied pedagogical agents can provide an effective environment for experientially grounded learning. However, such learning environments to date have been confined to one agent and one user. While a single agent single user setting simplifies interaction modeling, the richness of naturalistic multiparty interaction is severely compromised. In addition, the potential benefits of collaborative learning cannot be realized.

In this thesis, we analyze the different capabilities that agents need to possess to behave believably in the context of multiple users and multiple agents. A generic four-layer agent architecture with multiparty interaction support is introduced to address the challenges that arise in agent planning and task execution, communication and understanding, as well as effective coaching of student learning. A Newtonian 3D learning environment for agents and users is presented to illustrate the effectiveness of the agent architecture. An evaluation was conducted to determine the naturalism of the multiparty interaction and the extent of improvement in student learning.

The approach we have adopted in constructing agents with multiparty interaction support can be regarded as a generic step towards addressing and solving issues related to effective student interaction and learning for a 3D virtual learning environment in any sophisticated domain of learning.

CHAPTER 1. INTRODUCTION

1.1 Overview

Immersive virtual worlds are increasingly favored as a computer-mediated channel for human interaction and communication. These worlds present a rich and interactive environment for users to engage in. They can act on objects in the world as well as interact and converse with one another. Realistic three-dimensional representations of other users in the world create an enhanced sense of social co-presence. Users can benefit when such environments are augmented with believable virtual agents [1] [2]. For instance, they can be aided in task performance in a very natural social way. In the domain of education, several well known pedagogical agents have been developed [3] [4]. Most of these agents operate within a one-to-one tutoring scenario, and their effectiveness has been well demonstrated [5]. User learning gains in such dedicated tutoring settings are usually superior to what is achieved using traditional one-to-many teaching in the real world. Technology creates opportunities for innovation in pursuit of supporting computer-mediated forms of collaborative learning. It is possible to create multi-agent single user as well as multi-agent multi-user learning environments, thus fostering student learning in a more social setting. The inclusion of multiple agents allows the designer of a learning environment to engender multiple approaches to solving a problem and to appreciate multiple, often diverse, perspectives on an issue. However, several challenges arise when we seek to enlarge the interaction space to one that includes multiple users and multiple agents. First, the functional role of each agent needs to be carefully designed so as achieve complementarity with just the right amount of overlap and redundancy. Second, interaction between all participants in the learning environment, both real and virtual, must be intelligently handled so that learning and coaching processes unfold in a natural and effective manner. Third, the modeling of student learning needs to be characterized and managed at both the level of the individual as well as that of the group. A flexible agent architecture is essential to create a virtual world learning environment that responds dynamically to the situation faced "on the ground."

In designing the pedagogical function, we can draw from previous work that advocates the desirable characteristics of a good intelligent tutoring system as one that should be able to (1) flexibly plan the learning process, (2) detect and correct student misconceptions and errors, (3) improve students' critical thinking ability, and (4) provide personalized coaching by responsive adaptation to the changing requirements of users over time. Early tutoring systems often restrict the actions of users so as to achieve a high level of learning effectiveness, based on the system designer's concept of "correct" learning. However, the learning outcomes that can be achieved using such systems are today regarded as being stylized and overly restrictive on users' actions and commission of error.

1.2 Research Objectives

Creating an effective multi-agent collaborative learning system is the primary goal of the research. We decompose this high-level objective into two key elements.

First, on the technology aspect, this research gives us opportunity to explore approaches of integrating multiple embodied agents in a virtual environment. It imposes on us the challenges not only to incorporate an appropriate protocol for multi-agent communication, but also to enhance the agents' social intelligence of behaving believably in front of multiple human users and other computer simulated agents.

Second, we also aspire to boost the effectiveness of the learning facilitative process utilizing the technology we embrace. Using multiple instances of agents undoubtedly gives rise to more research interests compared to a single agent approach, however, the effectiveness and efficiency of multiple agents in a learning application cannot be taken for granted. Therefore, the real challenge for us will emerge when we try to combine the technology and education seamlessly and effectively. Of course, a well established preliminary understanding of the student learning problems is indispensable to the successful fulfillment of this learning objective.

In short, this research intends to strike an appropriate balance between creativity of the technology use and the effectiveness of the technology so used.

1.3 A Multi-Agent Virtual Physics Learning Environment

In the multi-agent system that we develop, we use Newtonian physics as the learning domain and natural language (both spoken and typed), mouse manipulation etc. as the form of human-computer interaction. Prior research has revealed that fundamental misconceptions relating to Newtonian physics are deeply-entrenched and widespread. It has proven to be difficult to shift such misunderstandings because of the strong interplay between knowledge, experience, and beliefs. The use of natural language as the basis of interaction between users and machines has the advantages of naturalness and enhanced ease of communication. However, making sense of the goals, intentions, and beliefs of students is hard.

The agents in the learning environment should facilitate student learning. The transfer of learning should be sufficiently smooth so that students can benefit from the interaction with the agents as well as other users.

To concretize our idea, we have devised a virtual spaceship environment for agents and users to cohabit. Three agents with assorted functional roles have been constructed. Ivan, the instructor agent, takes charge of describing the tasks for users. His duties also include resolving students' doubts relating to the procedures of learning task execution. Ella, the evaluator agent, judges users' utterances and provides feedbacks accordingly. She has the expertise of identifying, classifying, and correcting users' misconceptions. A set of strategies are implemented by her whenever an individual user or a group of users have exhibited certain misunderstanding towards the knowledge through their activities in the virtual environment. The third agent, named Tae, is a thinking helper agent. He initiates and mediates the conflicts among the students repeatedly to help them to collaboratively identify and overcome learning impasses. The students' understanding could often been improved by such a reciprocal evaluation.

1.4 Structure of the Thesis

This thesis will first present the design of a multi-agent, multi-user learning environment for studying Newtonian physics. In the later part, the system illustration and user study will be also presented. The entire thesis comprises nine chapters.

Chapter 1, **Introduction**, gives an overview of the motivations and objectives that this thesis aims to achieve.

Chapter 2, **Literature Review**, discusses the various research areas that ground this multi-disciplinary project. Relevant reviews cover the knowledge of human computer interaction, education effectiveness, collaborative desktop VR learning and agent technologies.

Chapter 3, **Intelligent Agent Architecture**, presents a generic four-layered architecture for supporting agents' behaviors in a multiparty learning environment. The construction of such architecture and the interaction among the system components is described.

Chapter 4, **Task Oriented Multiparty Interaction**, illustrates the system flow by presenting a task oriented approach. It also depicts an interaction model to regulate the collaborative activities among agents and users on a high level control. The issue of turn taking decisions will also be address.

Chapter 5, **Understanding and Responding**, clarifies the emerging interpreting challenges due to the increase of agents and users in a virtual environment. Four sub-components, namely, speech act classifier, ambiguity resolver, intention capturer, and behavior analyzer, are introduced to enhance agents' understanding ability.

Chapter 6, **Pedagogical Function**, elucidates the design of the agents' functional roles and the concept of the techniques that agents use to help users improve their knowledge and understanding.

Chapter 7, **System Framework and Illustration**, reviews the example scenarios in our virtual physics learning environment. It also explicates how agents cooperate to behave intelligently in order to foster an effective learning environment for multiple students.

Chapter 8, **Evaluation**, describes the evaluation methodology and observed results of the user study performed on the virtual physics learning environment.

Chapter 9, **Conclusion**, summarizes the thesis and states our achievements and contribution. Possible future work is also discussed.

CHAPTER 2. LITERATURE REVIEW

Three-dimensional virtual environments have become increasingly popular as a form of interactive technology because its application often emerges in the fields of e-commerce, military, medicine, entertainment and education. Together with intelligent agent technology, it certainly will make big influence to our life. In this chapter, a literature study on animated pedagogical agents and virtual learning environment will be presented.

2.1 Background

Developing a virtual learning environment integrated with intelligent pedagogical agents requires a lot of preparations. Five years ago, a research system named C-VISions [6] had already been developed in the Computer Science Department of the National University of Singapore. The C–VISions learning environment is modeled as a set of interconnected virtual environments. Each virtual world contains its unique scenarios for learners to participate in. Multiple users could not only use the audio or text chat features to communicate with each other, but also manipulate the virtual objects so as to fulfill their learning tasks. (See Figure 1)

This early version of C-VISion system can be regarded as a pragmatic step towards implementing the Experiential Learning Cycle (see Figure 2) proposed by Kolb [7]. Active experimentation yields concrete experience that provides the basis for reflective observation which eventually leads to abstract conceptualization, and the cycle iterates. In the process, students' understandings are transformed both extensionally and intentionally while comprehension is grounded in apprehension.

Figure 1. C-VISions virtual learning environment

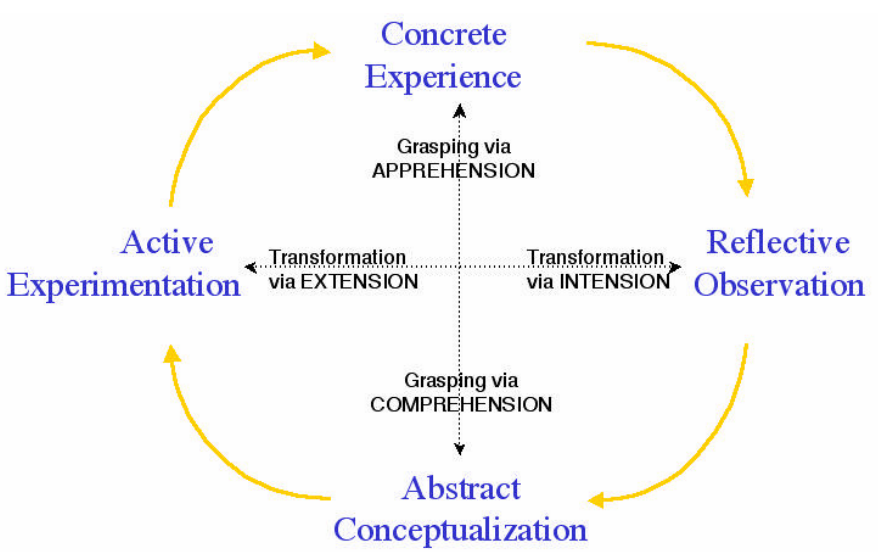


Figure 2. Kolb's experiential learning cycle

Nevertheless, along with a series of user empirical studies conducted, we realize the barriers occurring during student passive learning are not easily overcome by the mere presence of the virtual environment [8]. Learning impasses can arise when students, interacting in a 3D virtual world environment, are unable to make further learning progress on their own. This kind of situation may occur either when group members do not possess the requisite knowledge needed to bootstrap themselves out of their predicament during the learning process or when all group members mistakenly believe that their incorrect conceptual understanding of a science phenomenon is correct. This weakness motivates us to transform the virtual environment towards an agent enhanced learning setting.

As one of the pioneer embodied agent research work in NUS, a virtual agent, Elva, [2] was developed and incorporated with the C-VISions virtual world framework one year ago. Elva appears as a tour guide for a virtual art gallery. Whenever a user enters the room, she will start to carry out her tasks -- to guide the users walking through different sculptures based on a simple planning system. The intelligence of the agent enhances the power of the system. There are two aspects. First, Elva is able to answer a user's queries using a natural language format based on Speech Act Theory [9] . This feature increases the richness of interactions as well as the realism of the virtual environment. Second, Elva's planning system grants users the flexibility of visiting the gallery on his/her own, i.e. for the active users, Elva will just accompany instead of lead the tour.

Figure 3. Elva: An embodied tour guide agent in a virtual art gallery

Although few learning related features have been built into Elva, this system could still be regarded as the Lab's first successful attempt to integrate the agent technology into the virtual environment. What's more, the experiences we have gained during the development of Elva have revealed some possible future directions for us to improve on.

Multiple Users

Elva's virtual world is confined to one user, albeit the networking infrastructure of C-VISions can support multiple users. The weakness is ascribed to the lack of Elva's social intelligence to confront two users or visitors simultaneously. For example, if Elva is presenting an artifact to one visitor, she does not know how to entertain the second joining visitor according to common social customs. Additionally, this problem becomes essential in a learning environment since collaborations among learners are always vital.

Multiple Agents

The benefit of putting Elva into the virtual environment is unassailable. It greatly enhances the user experience but raises one interesting question. What if there are more agents? Although it seems a little awkward in a real life to have two guides in a museum or two teachers in a classroom, the multi-agent approach in a virtual environment definitely benefits our life and we will get used to it sooner or later. In a real world, the human resource has to be limited due to cost but it becomes negligible in a virtual world. This is why we can create as many agents as possible, provided that they add useful value to the virtual environment. Besides, we realize most current virtual world systems focus on the interaction between a single agent and the user which often cannot reflect the richness of the interaction in a real social environment. This could be another attractive reason to motivate us to explore the possibility of integrating more than one agent in the virtual environment.

Interaction Model

Once there are multiple users and multiple agents working on a learning problem cooperatively, shared social interaction can serve an instructional purpose. However, when the number of participants (either users, agents or mixed) increases, the overall interaction in the virtual environment becomes intricate and difficult to manage. Without proper management, some combinations of turn regulation settings during interaction may lead the learning process astray. For example, free-form interactions can help users to become engaged in the virtual learning experience, but it can also make them puzzled about the underlying learning goals and processes due to a lack of guidance. On the other hand, if the system restricts the variety of possible interactions, users' learning flexibility is lost. These considerations make clear that it is crucial to implement an effective interaction model in a multi-agent multi-user virtual learning world.

Natural Language Interpretation

Elva's competency of natural language understanding is achieved by the use of Speech Act Theory. This theory claims that every user's expression can be mapped to a certain intention. Based on this idea, agents could virtually give a meaningful answer to any user utterance due to the limited number of intentions under a certain knowledge domain. This approach of natural understanding has become popular in developing embodied agents [10] because of the factuality of implementation. However, there are also side effects due to the simplification of the theory. First, Elva only considers user's latest expression and disregards any historical context. Second, Elva cannot monitor multiple users' discussion. These findings provide us with big challenges to enhance the agent's natural language interpretation ability in a multiple users environment.

2.2 Reviews of Related Systems and Technology

This section reviews different important design considerations and evaluates the suitability of the related approaches.

2.2.1 User Intention Interpretation

Artificial Intelligence is a fundamental support for creating lifelike agents. Here we will examine the several approaches of simulating agent's intelligence to understand users' verbal and non-verbal behaviors.

For interpreting users' verbal expression, Seung [11] has pioneered an appeal using finite state machine for classifying the speech acts. Dialog acts are identified by automata which accept sequences of keywords defined for each of the dialog acts. Pattern matching techniques are applied for matching the queries with responses. This approach illustrates a simple and clear solution to classify the speech acts, hence extract user's intention. Nevertheless, the lack of reference resolution [12] and the use of predefined responses limit the agent's response ability.

As an effective supplementary channel, non-verbal user behaviors are also crucial for agents to analyze users' intentions. Rea [1] is an embodied, multimodal real-time conversational interface agent that acts as a real estate salesperson. It is equipped with a user behavior recognizer and classifies user's gestures as they occur. The classification is based on Hidden Markov Model (HMM) which categorizes a user's non-verbal behavior into one of the seven intentions based on a large offline training set.

2.2.2 Multiparty Interaction

Multiparty interaction in a virtual environment refers to the activities or conversations shared by three or more than three persons. It differs from one-to-one interaction significantly due to the complexity incurred by the quantitative increment of the participators. A superior modeling of the interaction among multiple agents and users should be constructed to offer a realistic learning environment to the students.

The concept of transition relevance places (TRP) was proposed by Sacks [13] to address turn taking issues in a multiparty environment. The TRP points refer to the moments when a speaker's discourse has natural points for others to begin their turns. Padilha [14] [15] continues the TRP topic by discussing the attributes of turn taking behaviors and suggests a list of possible events signals for TRP to occur.

Mission Rehearsal Exercise project [16] contains an interactive peacekeeping scenario with sergeant, mother and medic in foreground. A set of interaction layers for multiparty interaction control regarding contact, attention, conversation, social commitments and negotiation are defined. Furthermore, in the conversation layer, components such as participants, turn, initiative, grounding, topic and rhetorical are defined to build the computational model for social interaction customs. This facilitates the management of the multiparty dialog.

Various considerations for multiparty including the idea of defining group interaction pattern are discussed by Dignum [17]. This concept of interaction pattern is carried forward by Suh [18] when she proposed a taxonomy of interaction patterns for a tutoring scenario.

2.2.3 Discourse Management

A virtual animated agent often needs to show, explain, and verbally comment on the environment, users' behavior or triggered events. This requires the agent to effectively organize his dialog in a clear structure. We denote this knowledge as an agent's competency of discourse management.

Personalized plan based presenter [19] (PPP-persona) generates discourse behaviors according to a predefined script which is also affected by the agents' self behaviors in real time. A presentation script specifies the presentation acts to be carried out as well as their temporal coordination. Self behavior comprises not only requisite gestures to execute the script but also the navigation acts, idle time gestures and immediate reactions to occurring events in the user interface. The novelty of PPP is that the presentation scripts for the characters and the hyperlinks between the single presentation parts are not stored in advance but generated automatically from the pre-authored document fragments and items stored in a knowledge base.

Herman [20], an animated agent that helps user to learn how to "Design-A-Plant", monitors students as they assemble plants and intervenes to provide explanations about botanical anatomy and physiology when they reach an impasse. The explanation process is separated into two levels of reasons. The surface reason is to provide problem solving advice, and the deeper reason is to provide students with a clear conceptual understanding in the domain.

Rickel and Lewis have developed Steve [21], a pedagogical agent as shown in Figure 4, to teach the operations of maneuvering a submarine. Steve can conduct training for students through demonstration, monitor and explanation. A hierarchical approach has been adopted for clarifying tasks. Different steps in a plan have been defined as nodes in the task hierarchical tree. Ordering constraints and casual links indicate the relation among steps and pre-post conditions respectively. Whenever Steve needs to explain the purpose of certain task step to the student, the pre-post conditions are used to help him to trace the reasons as well as organize the dialog discourse.

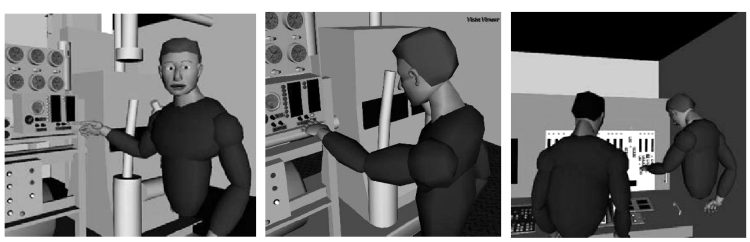


Figure 4. Steve - an intelligent pedagogical agent

2.2.4 Intelligent Tutoring System and Related Concept

In a broad sense, a multiparty virtual learning environment can be regarded as a form of intelligent tutoring system (ITS). Many of the early ITSs unveil the essential features of a teaching and instructional system.

El-Sheikh [22] models an intelligent tutoring system in term of four components: expert model containing cognitive knowledge and solution strategies in a particular domain; student model describing the student understanding status; pedagogical module to control and influence the learning process; and communication module in charge of interaction with the student.

Teaching style has been indicated as one of the important keys to produce a good tutoring system [23]. The traditional *testing* style only gives student correct or incorrect answers without additional explanation. Other systems adopt a *telling* style, which is a style usually happening in a traditional lecture. Virtual agent keeps conveying correct or incorrect information to users. *Coaching* style requires agents to act like a teacher to correct student error by explanation or suggestion. Learning environment styles permitted user to create the problem for learning. Different state of the problem can be tried out and agent will give assistance only at suitable time.

Experiential learning [7] can apply to students learning in the virtual environment through experience. Experiential learning is often used by providers of training or education to refer to a structured learning sequence which is guided by a cyclical model of experiential learning. Less contrived forms of experiential learning (including accidental or unintentional learning) are usually described in more everyday language such as 'learning from experience' or 'learning through experience'.

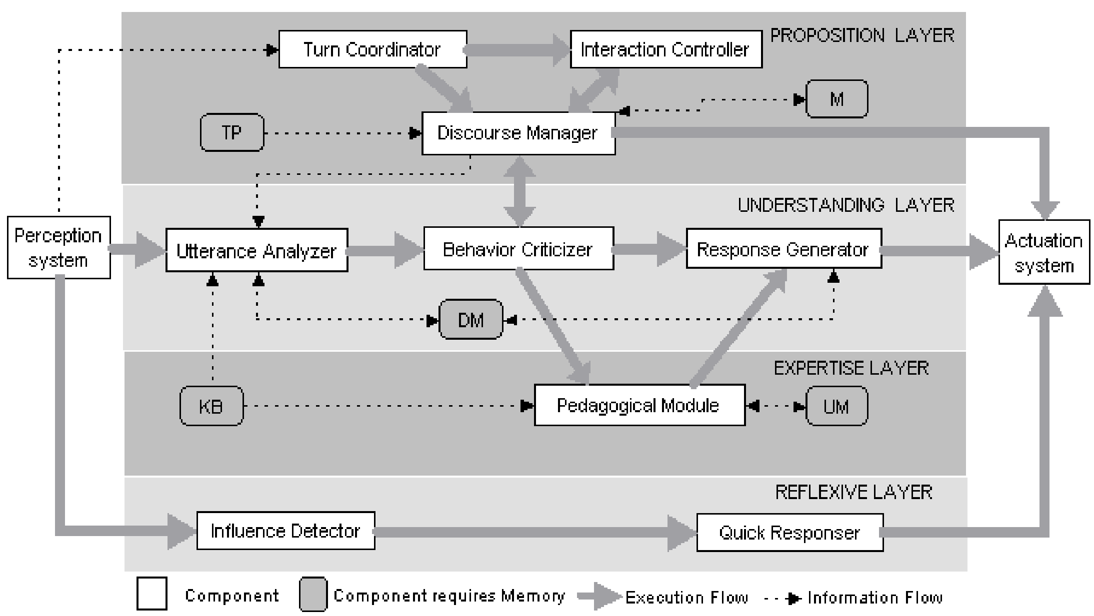
The design of learning task also plays a vital role. Herman the bug [20] adopts a style of learning by construction. Student may combine different components such as root or stem to form a plant. Steve [3] allows user to monitor the sequential steps of a demo, followed by practicing, and questioning. The WhizLow agent [24] inhabiting the CPU City 3D learning environment depicts the location information within a CPU through navigating. WhizLow agent uses a misconception detector, classifier and corrector to help users improve understanding.

CHAPTER 3. INTELLIGENT AGENT ARCHITECTURE

Our agent's behavior is determined by the considerations of general task execution, group multiparty interaction and self multimodal animation. Therefore, a well designed agent architecture must be realized that enables the agent's multitasking ability in an effective and efficient way.

3.1 Overview of the Agent Architecture

An agent is intelligent by virtue of its ability to acquire and apply knowledge. We have designed a four-layer agent architecture for this purpose (see Figure 5). From top to bottom, these layers achieve the agent intelligence in terms of task fulfilling, social communication, pedagogical intelligence and adaptive ability.



TP: Task planner, **M**: Memory, **DM**: Dialog Model, **KB**: Knowledge Base, **UM**: User Model

Figure 5. Four layer intelligent agent architecture

The perception system input component in the agent architecture constantly updates the surrounding environment information for the agent to make the right decision. It enables the agent to "see" users' movements as well as "hear" group conversations.

On the output side, the actuation system, in conjunction with the knowledge base, handles the agent's animated behaviors and generated responses. Synchronization has been implemented to coordinate the timing of different animated channels such as body posture, facial expression and locomotion. The actuation system is also powered by the AT&T text to speech voice engine. It endows the agent with the ability to produce the realistic human voice utterance.

3.2 Four Layer Agent Architecture

The fours layers in the agent architecture, namely, proposition layer, understanding layer, expertise layer and reflexive layer are implemented in a multiple threads manner. They process autonomously as well as influence each other's execution.

The *proposition layer* determines the way the agent carries out its task. A *task planner* first assigns the agent a task then passes control to the *discourse manager*. The *discourse manager* then decides the agent's role for the current task by referring to the agent's *memory* module. This role information helps the *discourse manager* determine an interaction pattern for the *interaction controller*. Different agent *interaction controllers* negotiate and synchronize a common interaction pattern. An interaction pattern is defined as a set of primitive interactive behaviors among agents and users in a dialog. The *discourse manager* serves as a bridge whenever the *interaction controller* needs to inform the *actuation system* for the multimodal behavior output. When the *discourse manager* detects any user behaviors conflicting with the current interaction pattern, the *interaction controller* pauses. As a result, a new session of the dialog is initiated by the user. The *turn coordinator* is then invoked to help the agent decide turn taking requests during the conversation.

The *understanding layer* helps the agent determine the user's intention. The *utterance analyzer* tracks a user's intention via four modules: (1) a *speech act classifier* categorizes the user's speech; (2) an *ambiguity resolver* tries to achieve grounding in a dialog by cooperating with a *dialog model* which memorizes and manages all the dialog states; (3) an *intention capturer* differentiates between listeners' roles and identifies the implicit intention in a speech act; (4) a *behavior analyzer* infers the user's intention by referring to a series of previous actions. The *discourse manager* always passes the current task information to the *utterance analyzer* for further interpretation. The *utterance analyzer* transfers the determined utterance to the *behavior criticizer* to identify user misconceptions or errors. Finally, the *response generator* engenders a response and consequently the system control has been passed to the *actuation system.*

The *expertise layer* endows the agent with pedagogical intelligence. The *behavior criticizer* classifies user problems into errors, misconceptions, or thinking difficulties and passes the result to the *pedagogical module.* When that's finished, different agents with their respective pedagogical abilities solve the user's problems with the aid of a *user model*. The *user model*, as a reference database, maintains each individual's learning status. The *pedagogical module* passes control to the *response generator* when feedback is required.

The *reflexive layer* provides the agent with the capacity for quick, adaptive behavior. The *influence detector* helps the agent to make decisions related to joining or leaving a nearby dialog group with the location information perceived from the environment. The *quick responser* enables the agent to gaze at or walk toward moving users to achieve high social believability.

3.3 Multiparty Interaction Support

Focusing on multiparty interaction, the entire system can be visualized as a combination of different interaction levels (see Figure 6).

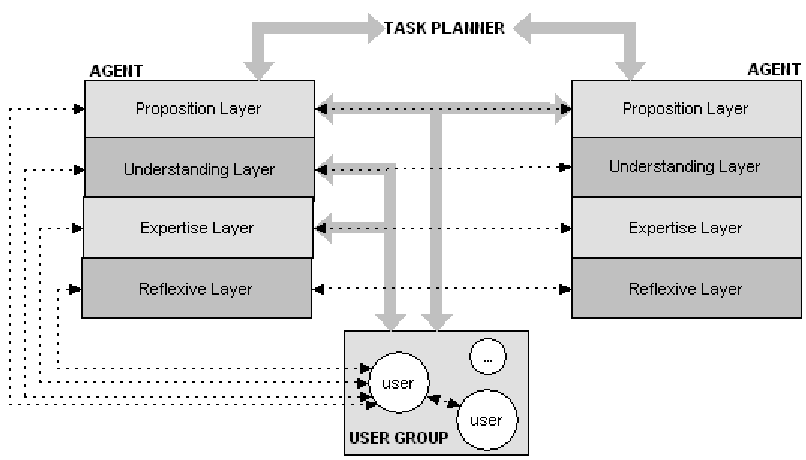


Figure 6. System view of multiparty interaction

Visualization of the entire system interaction enables us to scrutinize the behaviors among layers from different agents and observe how the agent deals with a multiparty situation. By lists all the possible quantity relationship between agents and users in the virtual environment, we are especially interested in the following classification of the interaction: single agent user interaction, single agent multiple user interaction, multiple agent interaction, and multiple agent multiple user interaction. The next section explains the detail how these interaction modes are realized in our system.

Reflexive behavior is always realized in a one-to-one interaction, either between two agents or between single agent and the user. The understanding process occurs at either the individual user level or the group level. Single user understanding is still the dominant activity for agents in the learning environment. Nevertheless, when the agent feels it necessary to analyze the behaviors for an entire group of users, the understanding layer will make use of the dialog model to achieve precise interpretation for the user group. The agent's *pedagogical module* also functions in both single user and multiple users' perspective. The agent corrects common misconceptions for each individual user and keeps those successful strategies for subsequent interaction. Regarding the task execution, the *task planner* serves as a coordinator for multiple agents to converge on a common execution plan through multi-agent communication. The *discourse manager* and *interaction controller* always keep track of the information from all the agents and users interaction to decide the interaction pattern for the entire group multiparty interacion. Similarly, turn taking is realized as a multiparty interaction because it requires continuous negotiation among multiple agents whose decisions are also influenced by the users indirectly.

3.4 Summary

This chapter introduces our four layer intelligent agent architecture. The four layers are disposition layer, understanding layer, expertise layer and reflexive layer. They address different issues concerning multiparty learning interaction in their respective dimensions. Besides, a system level visualization is also presented to explain how different types of interactions take place in our virtual environment.

CHAPTER 4. UNDERSTANDING AND RESPONDING

Natural language permits rich communication to take place between machines and users, but it is always one of the most complicated problems in computer science. This chapter describes how the agent interprets a user's utterance by analyzing both verbal and non-verbal user behaviors and agent understanding in the context of multiple users.

4.1 Utterance Analysis

Utterance analysis is divided into four modules: (1) speech act classifier, (2) ambiguity resolver, (3) intention capturer, and (4) behavior analyzer.

Speech Act Classifier

The *speech act classifier* adopts the pattern matching technique to identify a user's intention. In the preparation phase, word stemming, reference resolution, stop word removal, synonym replacement and keyword extraction are applied to facilitate information processing. Next, the *speech act classifier* attempts to use a finite state machine to identify the pattern of an input sentence. Once the pattern is extracted successfully, a *pattern--speech act* mapping table is consulted for transforming the pattern into a user speech act defined especially for our learning environment (see Table 1). It is not uncommon that different sentence patterns may lead to the same speech act. This many-to-one relationship significantly minimizes our efforts to capture the intention of the unlimited possibility of user's utterance. Consider the following illustration. The patterns of "why", "what causes", and "what is the reason" could be mapped to the same speech act named "question_why". At the end of the speech act classification procedure, the user's utterance can be represented as a combination of a speech act and several keywords.

Categories	**Speech Acts**
Commission	Think, Guess, Compare..
Question	Why, When, Who, Where, YesNo..
Expression	Greet, ISee, interest, Sad, Afraid, Giveup, Improve…
Request	Explain, Description, Repeat, Demo, Clarify, Suggest…
Declaration	Comment, Puzzle, Summary, Conclude, Agree, Disagree…

Table 1. Speech act classification

Ambiguity Resolver

The *ambiguity resolver* improves interpretation when the reference in a dialog cannot be figured out by the agent during the preparation steps of speech act classification. Names and locations are some of the potential candidates for creating ambiguity. The *ambiguity resolver* informs the predicament to the *dialog model* so that the latter can notify the *response generator* to issue a verbal request for the speaker to rephrase his utterance. Once the ambiguity is resolved, the speech act classification procedure is carried out as usual.

Intention Capturer

The *intention capturer* probes the user expression and discovers inconspicuous information such as implicit requests for action or the information related to listeners' roles.

A verbal response from the agent is not always sufficient to entertain a user's request. Some users' utterances express the intention for an action instead, and some request both. For instance, the question "can you do a demo for me?" not only requests a verbal agreement "yes", but also a real action of "demo". Our system integrates two methods to identify these implicit requests. First, the agent uses predefined templates to match the user's utterance to an implicit action. Second, the agent is capable of reading the user intention through an analysis of the user's previous behaviors through the *behavior analyzer* (discussed in the next paragraph).

To determine the listeners' role from an utterance is also a complicated process in a multiparty environment. Unlike a one-to-one interaction which always assumes the listener as the requested action performer, in a multiparty environment, an intention like "A requests B to inform C to ask D to do something" leads to sequential chained consequences, and every participating agent has to perform the requisite actions in a timely fashion. A recursive approach is adopted here to separate the header ("A request" in the example) and encapsulate the remaining requests as a whole for the next participator agent ("B" in the example) to proceed.

Behavior Analyzer

The *behavior analyzer* classifies the user's intention by focusing on the sequence of the user's past behaviors. It stores the recent behaviors for each user and compares them with the supervised offline user testing data in order to classify the user's intention. The result from the *behavior analyzer* often assists the *intention capturer* to interpret the implicit requests from user's actions.

4.2 Multi-party Dialog Management

The *dialog model* manages the responses from different users in a multiparty environment.

For an individual participator involved in the current conversational group, the *dialog model* maintains an individual dialog state which records the last few utterances. They are saved for future referencing.

At the group level, the *dialog model* maintains a *response pool* to store every pending response in a timely fashion. This effectively addresses the problems that arise when multiple users express their utterances continuously one after another before the agent has the chance to become a speaker to reply. A pruning step is applied to remove any redundancies or conflicts among the responses in the *response pool* before the agent speaks.

The *dialog model* also recognizes the utterance or intention of a group. Group interaction modes such as "discussion" and "debate" have been defined to categorize group behaviors. The agent's *discourse manager* scrutinizes this group interaction information to analyze the accurate interaction pattern among multiple users.

4.3 Summary

This chapter illustrates different agent components for enhancing its interpretation ability. *Speech Act classifier* categorizes user's interaction; *ambiguity resolver* filters the uncertainty in user's utterance; intention capturer further analyzes user's implicit intention; *behavior analyzer* helps agent to produce deliberative decision based on the sequence of users' non-verbal behavior. In addition, *Dialog model* enhances agent's interpreting ability in a multiparty environment by storing the conversational data under both individual and group schemes.

CHAPTER 5. TASK-ORIENTED MULTIPARTY INTERACTION

Our design of the task-oriented and mixed-initiative multiparty interaction is based on a sophisticated structure. This structure allows agents and users to flexibly execute tasks efficiently. It also deals with the situation when unexpected user behaviors occur.

5.1 Task Execution

Task execution is made flexible through a graph structure implementation (see Figure 7). Each rounded rectangle denotes a group of several tasks. The arrows indicate the ordering constraints among the tasks and the groups of tasks. The task planner sequentially picks a group when executing tasks. A single task can be compulsory or optional depending on the ordering constraints. For example, at *B*, task 2 and task 3 are both compulsory but the execution ordering between them is flexible. At *C*, finishing either task 4 or task 5 is sufficient to proceed to the next group of tasks. At *D*, task 7 contains a superset knowledge over task 6, hence, finishing task 7 is adequate to advance without task 6 but not vice versa.

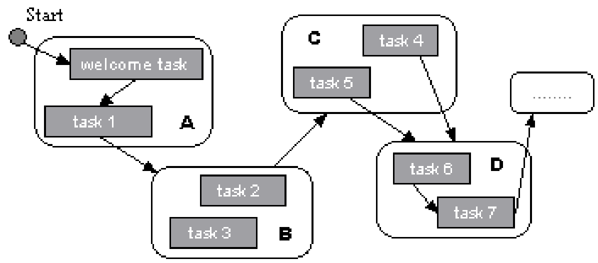


Figure 7. Illustration of task planning

5.1.1 Task Structure and Terminology

Each task is designed in terms of a three layered topology comprising: (1) *topic layer*, (2) *interaction function layer*, and (3) *interaction pattern layer*. The *topic layer* consists of the task description, the conditions for achieving the different stages of the task, the ordering constraints with other tasks, the procedure information such as what tools are used during the task, and some common misconceptions about Newtonian laws. The *interaction function* denotes the high level pedagogical techniques, such as "explanation" or "demo", which are usually defined as some complex tasks in a tutoring domain. The *interaction pattern* describes basic turn taking information for multiparty scenarios. Fifteen interaction patterns have been defined for our tutoring scenario (see Table 2).

Interaction Categories	**Interaction Patterns**
Social	Initiate topic, Invite User, Leave topic, Terminate topic and Greet
Understanding	Provide information, Q&A, Knowledge linking, Comparing theorem
Collaboration	Integration, Agreement, Suggestion
Miscellaneous	Disagree, Illustrating
Supervising	Give feedback

Table 2. Interaction patterns

Figure 8 shows a flow diagram for an interaction pattern called “knowledge linking”. The agent initiates the interaction by describing two related problems, followed by either a group discussion or a single user’s conclusion. This interaction pattern finally ends with some feedback given by the agent. The benefit of having such an interaction pattern is to construct an optimum model so as to achieve the efficiency and effectiveness for students learning in a multiparty environment.

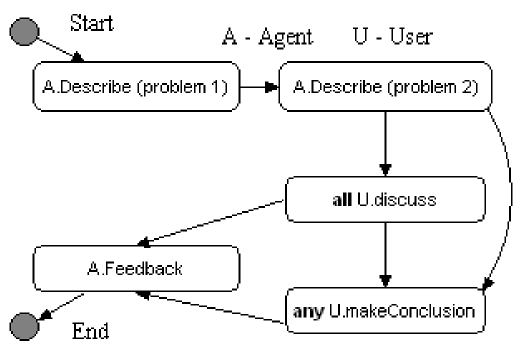


Figure 8. The interaction pattern of “knowledge linking”

5.1.2 Cooperation of Task System Components

Task execution follows the terminal nodes of the hierarchical tree with the ordering constraints. A terminal node is either an *interaction function* or an *interaction pattern* (see Figure 9). The content of the lower layer node is partially determined by its upper layer node. For example, to execute an *interaction pattern* called “provide information”, the *interaction pattern* retrieves the description from its parent node, which is an *interaction function* called “demo”. “demo” then references its own parent node for retrieving further elaborated interaction information. In this example, the *interaction pattern* designs the way to “provide information”. It informs agents what the desired turn taking behaviors are so that the agents can evaluate users' as well as other agents' behaviors. The *interaction function* "demo" restricts the type of the information to provide so that the interaction pattern only provides information relating to a demo such as the steps needed to execute the demo. Sitting on the top level, the *topic layer* determines the detailed content of the information such as "which demo should be illustrated".

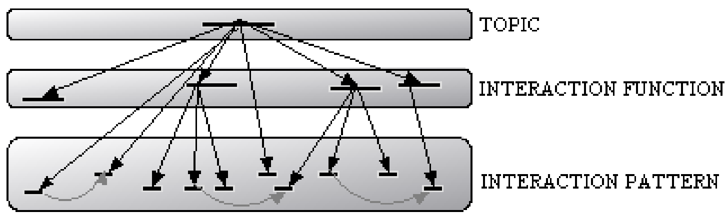


Figure 9. Hierarchical task topology

5.1.3 Rules for Applying Interaction Models

In a virtual environment, all interaction patterns are initialized by the agent. An interaction pattern is usually triggered according to the task description, but sometimes it is also invoked when the agent notices that the pre-conditions of the interaction pattern have been met. When the agent starts executing an interaction pattern, all users and other agents' behaviors will be recorded and analyzed for pattern retrieval. Once all the requisite behaviors are performed in the sequential order requested in the interaction pattern, the interaction pattern is considered terminated. Further explanations about the agents' rules for applying interaction pattern are given in section 6.2.

5.2 Turn Taking in Multiparty Conversation

As stated, the *turn coordinator* activates when unexpected user behaviors occur. With the *turn coordinator*, every agent can express his turn request at his will if the conversation does not have an interaction pattern to follow. Agents compute their turn taking bidding scores carefully, and a final comparison occurred in the server will announce the result of turn bidding requests.

When there is a short silence or the content in the dialog indicates a speaker shift, the agent with the highest turn bidding score will be the next speaker.

The turn bidding score is computed as $m * t * (a * f + b * d) * r$ where,

m: indicates the agent's name has been mentioned by the speaker. If yes, m = 1, otherwise m = 0.1,

t: the amount of time elapsed from the moment of the turn request until the moment when the turn scores are compared

f: the angle between the agent and the speaker's face orientation

d: distance between the speaker and the agent

r: importance level of the utterance the agent is going to articulate

a and b are the coefficient values for adjusting the importance of physical position during turn coordination.

5.3 Issues

Several potential problems concerning interaction models may arise during the system task execution: (1) how to identify user interaction type; (2) what if users do not follow the interaction pattern; (3) which agent to carry out an interaction pattern.

5.3.1 Identification of User Interaction Pattern

The *discourse manager* identifies user behavior with help from the *behavior criticizer*. The *behavior criticizer* receives both verbal and non-verbal user behavior information from the *utterance analyzer*. To recognize an interaction pattern from a user's non-verbal behavior, the *discourse manager* evaluates the environment state in order to analyze the effect of users' behavior. For user verbal behaviors, the *discourse manager* validates the group's intention through the *dialog model* which conserves the history of conversation for every user. However, if there is only a single user involved, the *discourse manager* analyzes the information from the *utterance analyzer* directly.

5.3.2 Dealing with Unexpected User Behaviors

There are three types of unexpected behaviors during the execution of an interaction pattern. First, the user behavior does not reveal sufficient information to be recognized as a form of valid behavior defined in an interaction pattern. In this case, the *discourse manager* informs the *dialog model* to request for elaboration. Second, the user behavior is distinctly contradictory to or irrelevant to the defined behavior in the interaction pattern. When the agent realizes this, it will not immediately "force" the user to behave according to the interaction pattern by issuing a command. Instead, the *turn coordinator* is employed to allow the agent to accommodate unpredictable turn settings in this new session of user initiated dialog. This process will not cease until the session of conversation exceeds a preset threshold or the user behaviors naturally become coherent with the interaction pattern again. Then the execution of the *interaction pattern* resumes. Third, users encounter difficulties in problem solving, or they display certain knowledge misconceptions. In this case, the *behavior criticizer* invokes the agent's relevant expertise/teaching modules. Once the user's difficulties have been solved thoroughly, the execution of the interaction pattern also resumes.

5.3.3 Selection of Agent to Initiate the Interaction Pattern

Since the description of an interaction pattern does not specify which agent to initiate the interaction pattern, two agents may compete to become the initiator. When this happens, the agent's *discourse manager* first determines the potential nearby competitors who are "free" at that time through the *perception system.* After exchanging the information of task execution priority, the agent with the highest score becomes the winner and initiates the interaction pattern. The initiator agent assigns the roles to the other agents, provided that the description in the interaction pattern requires the involvement of more than one agent. All agents also need to inform each other when they complete the current turn so as to achieve synchronization of their behaviors.

5.4 Agent Communication

The design of our interaction model helps virtual learning environment creators to model effective multiparty interactions. Multiple agents need to negotiate and inform each other of the interaction as it progresses. In Figure 10, the dotted horizontal lines denote the inter-agent communication, and the normal horizontal lines refer to the message or control passing within one agent. Different agents' *discourse managers* communicate with each other and also with the central task planner. The discourse manager at the same time sends and receives intra-agent communication messages to the *turn coordinator* and the *interaction controller* to coordinate the interaction. The *interaction controller* uses the *interaction pattern* information to decide and monitor the entire group's behaviors. Meanwhile, the *turn coordinator* also listens to the information from users and other agents' behaviors and decides the turn regulation settings without reference to an interaction pattern.

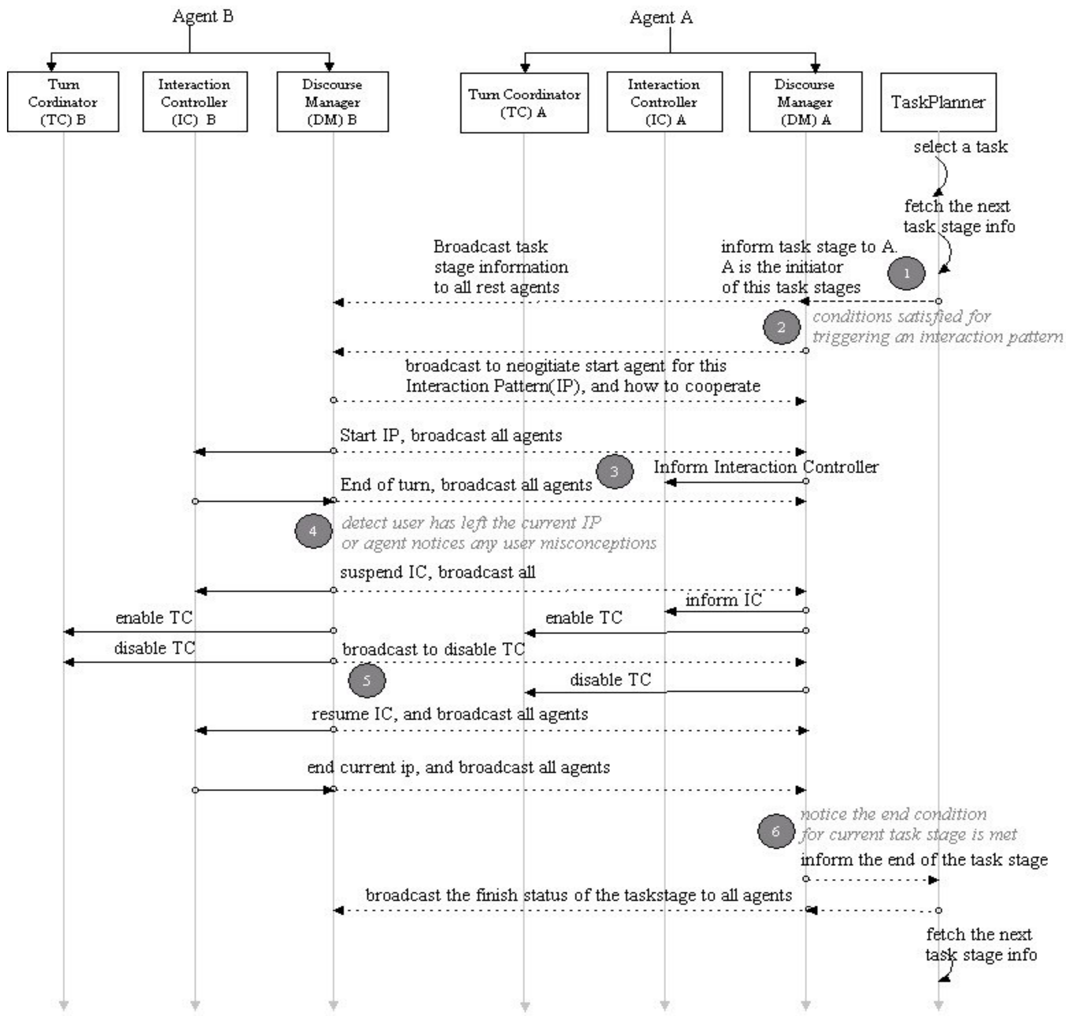


Figure 10. System flow of multi-agent communication

At time 1 (indicated as a grey circle in Figure 11), the *task planner* has just picked a task stage. It informs the details of the task stage to all agents' *discourse managers*. Agent A is selected as the initiator of this task stage according to the description defined in the task stage. At time 2, agent A notices certain conditions have been met to invoke an interaction pattern. It sends this information to Agent B's *discourse manager* to negotiate about which agent is going to be involved in this interaction pattern. At time 3, Agent B finishes its current turn actions defined in the interaction pattern. Its *discourse manager* sends this information to Agent A's discourse manager so that the latter knows the current status of the execution of the interaction pattern. At time 4, Agent B realizes that some unexpected user behavior has occurred. Its *discourse manager* suspends the running *interaction controller* and enables the *turn coordinator*. Agent A performs the same procedures after receiving the notification from Agent B. As a result, the interaction pattern is successfully stopped at the moment since all agents have synchronized their actions to pause the interaction pattern. At time 5, since the time exceeds a pre-defined threshold for user initiated conversation, agent B politely asks the users to perform some learning activities according to the interaction pattern. This step ensures that users do not spend time without any significant learning progress. When that's completed, agent B's *discourse manager* disables the turn coordinator and resumes the interaction controller. It also informs agent A's *discourse manager* to perform the same procedures. At time 6, agent A realizes the terminating conditions for the task stage have been met and informs the *task planner*. The *task planner* selects the next task stage after announcing the termination status of the previous task stage to all agents.

5.5 Summary

This chapter first elucidates the task structure defined and introduces the concept of interaction pattern as well as interaction function. Utilizing these structures, various examples have demonstrated how agents regulate the task flow and manage the interaction in a multiparty environment. Further discussion involves different issues which arise during interaction and how the turn coordinator intervenes. In addition, how the multi-agent communication supports the agent's task execution activities is also illustrated.

CHAPTER 6. PEDAGOGICAL FUNCTION

The pedagogical capability of agents is customized to fulfill the goal of effective teaching. Three embodied agents with different functional roles in our system are introduced. We will illustrate how agents promote user collaboration activities and how they identify users' problems and improve their domain understandings.

6.1 Design of Pedagogical Functions

Three agents with different functional roles cohabit our virtual learning environment. All agents utilize the same architecture. They differ only in respect of their distinct *pedagogical modules*, different priorities for task execution and unique rules to initiate the interaction pattern. The modularized design makes it easy to implement characteristic pedagogical agents on top of the existing agent architecture. In our system, both the instruction and evaluation agent are equipped with a pedagogical module which supports misconception correction. The evaluation agent Ella has a higher priority in helping students overcome misconceptions while the instruction agent Ivan has a higher preference for describing tasks and giving instructions. The third agent, the conceptual thinking agent, named Tae, provides scaffolding when users are not able to engage in critical thinking on their own. Inspired by UC Berkeley's Thinker Tools [6], Tae enhances users' thinking ability through the use of questions, hypotheses formation, investigation, and evaluation activities. As soon as user behavior reveals that the user has difficulties in continuing a task, the thinking agent's *pedagogical module* is invoked to raise reflection questions for the user. If the user is still puzzled about the task, the thinking agent requests other agents for assistance hence creating a multi-agent tutoring process.

6.2 Agent's Heuristics

A set of heuristics possessed by each agent has been designed to facilitate collaborative and group learning behaviors across virtual worlds. These heuristics are defined in terms of rules. A sample of these agent heuristics is listed below to provide a sense of the collaboration and group facilitation knowledge possessed by the agents. The heuristics assume application of the principle of conceptual conflict by design where embodied pedagogical agents deliberately create situations of experientially grounded conflict that triggers students' cognitive dissonance which, in turn, requires resolution.

Ivan, the Instructor agent:

• IF detect that students lack a critical knowledge component THEN provide information on the missing knowledge component [Rule 1]

• IF requested by Evaluator agent to set up a conceptual conflict THEN choose an appropriate conflict task and provide the task information to students [Rule 2]

• IF detect that students are not in agreement THEN reiterate recent utterances reflecting disagreement (to highlight that a conceptual conflict exits) [Rule 3]

• IF students have just completed a task from different frames of reference THEN invite users to share their different experiences (to project the conflict into the shared conceptual space for negotiation) [Rule 4]

• IF a new student joins the group THEN ask a group member to share the current group goal with the new student [Rule 5]

Ella, the Evaluator agent:

• IF detect that students have converged to a shared misconception THEN request Instructor agent to set up a conceptual conflict for them to resolve [Rule 6]

• IF detect that students are not able to decompose the conceptual conflict task into manageable parts THEN provide advice on decomposing the task [Rule 7]

• IF identify one student with a misconception and other students disagreeing THEN ask other students to elaborate on the reasons for disagreeing [Rule 8]

• IF detect that one or more students lack prior knowledge already possessed by other students THEN ask one of the other students to articulate the knowledge that the peers lack [Rule 9]

• IF detect that students have drawn an incorrect conclusion after carrying out a correct procedure THEN ask them to re-perform or re-analyze the procedure and its outcomes [Rule 10]

• IF detect that students have made an error in constructing a model of the scientific phenomenon THEN provide specific feedback on the step that is erroneous [Rule 11]

Tae, the conceptual Thinking agent:

• IF conceptual conflict task has been set up by Instructor agent THEN ask students to state a hypothesis or explanation that resolves the conflict [Rule 12]

• IF detect that students are not in agreement THEN ask them to re-examine and reflect on what might be causing the disagreement [Rule 13]

• IF one student articulates his/her explanation THEN ask another student for his/her opinion on it. [Rule 14]

• IF one student asks a question THEN ask another student if he/she can answer the question. [Rule 15]

• IF students contribute different instances or examples of something THEN ask them if there exists a valid generalization and, if so, what it might be [Rule 16]

• IF detect that student dialog lacks conceptual coherence THEN ask students to engage in problem restatement [Rule 17]

At the implementation level, these heuristics are represented either as rules to trigger the interaction patterns, or predefined in the task description. Pattern matching techniques are applied here again to extract the underlying understanding of the users by referring to both the context of the conversation and the historical observation about users' activities.

6.3 Misconception Detection and Correction

The *behavior criticizer* detects whether a user's current action or utterance could lead to an error, a misconception, or reveal a difficulty in task solving. The following situations illustrate some of the typical scenarios triggering the further process of the agent' *behavior criticizer*:

IF by classifying the speech acts, the agent realizes the users…

1. do not agree with him/her
2. have made some declarations, comparisons or conclusions
3. realize a misconception by themselves
4. have missing steps in performing task
5. cannot draw a suitable conclusion after a long discussion

THEN the *behavior criticizer* will trigger the *pedagogical module*, and the misconception detection process launches.

The misconception identification process is based on the first order predicate calculus (FOPC) which is able to describe objects, relations, properties, and events for the Newtonian laws learning domain in logical expressions. Some terms are listed in Table 3.

Terms Type		Instances
Constant	Notion	Force, Velocity, Acceleration, Gravity, Mass, Impetus, buoyancy, Object…
	Status	static, moving, accelerating, rotating …
	Condition	Free-Friction, Vacuum…
Function	Property	SpeedOf, MassOf, acceleration Of …
	Numeric	Add, Minus, Times, Avg, Sum …
Predicate	Operation	Move, Drop, Hit, Rotate, Fly, Project, Stop, Turn, Break …
	Relation	Increase, Decrease, Inverse, Unrelated, Equal…
	Property	Object, Environment

Table 3. FOPC defined for Newtonian physics learning domain

An expression such as “Objects with different mass drop at the same speed rate in a vacuum condition” can be represented as:

$$\begin{aligned}&\forall x,y \in \mathit{Object}, \mathit{Enviroment(vacuum)} \wedge \mathit{Drop(x)} \wedge \mathit{Drop(y)} \wedge \\ &(\neg \mathit{Equal(massof(x),massOf(y))} \\ &\quad \Rightarrow \mathit{Equal(accelerationOf(x),accelerationOf(y)))}\end{aligned}$$

Initially, there are a few correct FOPC expressions defined for each task. They are used for validating the user’s utterance. A misconception is identified if the user’s utterance conflicts with the existing facts.

A pattern matching algorithm is used to transform a user’s utterance (if the user expresses a meaning completely) or both the agent and the user’s expression (in the case when a user gives an answer in response to an agent's question) to a FOPC expression. If the information extracted from the user's utterance is insufficient for this conversion, one of two approaches is adopted. The agent may choose to ask the user for a detailed explanation, or he may leave the current user's utterance for future processing. However, if the user's utterance is identified as a consistent expression with the correct FOPC expressions in the existing facts base, this utterance is regarded as the user's correct conceptual understanding of the current topic. Consequently, appropriate feedback can be given. Otherwise, the agent continues processing the user's utterance to determine whether the user possesses some element of misunderstanding.

However, the agent must first ensure that the user's behavior is not a careless mistake before it attempts to correct the misconception. It allows the user to re-evaluate his last utterance by asking him for a confirmation. If the user reasserts his incorrect answer, the agent then regards it as confirmation of a misconception. To make the correction procedure work, the agent has three strategies available: *Recall*, *Relate*, and *Reflect*.

Recall requires the agent to search for previous successful strategies when solving the same misconception for other users. *Relate* refers to the related discourse plan, example, or experiment which can be used to help the user refine the understanding. *Reflect* indicates the agent's request for the user to contemplate on his own misunderstanding. During a user's reflection, if the misconception is realized and corrected by the user himself, the entire correction process is completed. The corrected concept and the procedures are saved into the *user model*.

There may be occasions when the agent is unable to correct the user's misconception within the time allocated for one correction session. When this happens, the misconception is stored temporarily into the *user model* and retrieved when a future *similar* misconception is encountered. Similarity between two statements is calculated by comparing the keywords of the Newtonian physics.

The information in the *user model* is stored individually for each user. But the procedures for correcting misconception can be retrieved every time the agent interacts with other users. Overall statistics show the frequent of the misconceptions that occur. Questions concerning the "popular" misconceptions are raised by the agent more frequently as a strategy to test a user's knowledge and understanding.

6.4 The Design of Learning Tasks

Choosing Newtonian physics as our learning domain reduces our effort of acquiring the domain knowledge through the external experts. However, there is a stronger reason. Many people do not aware of their incorrect understandings about the Newtonian physics even after many years they studied the concept in the classroom. Therefore, there is a significant pragmatic value for us to develop a system under the domain of Newtonian physics.

Newton's three laws, with their concise forms of representation in textbook, never give students much trouble memorizing them or even writing them down in written test. Nevertheless, when students encounter real life problems, an alternative view of the physics relationships emerges. The source of such behaviors can be traced to deep-seated *naïve physical laws* students develop on the basis of everyday experience with the real world and this makes a deep understanding of Newton's laws very difficult to achieve.

Our learning design is based on Hestenes' Force Concept Inventory. The Inventory data provides a perspective of the widespread problem of commonsense misconceptions in introductory physics. The Force Concept Inventory is structured to require a choice between explanation based on Newtonian concepts and commonsense alternatives. The Newtonian force concept is broken down into six conceptual dimensions, all of which are required for deep understanding of the complete concept.

1. Kinematics
    a. velocity discriminated from position
    b. acceleration discriminated from velocity
    c. constant acceleration entails
        - parabolic orbit
        - changing speed
    d. vector addition of velocities
2. First Law
    a. with no force
        - velocity direction constant
        - speed constant
    b. with canceling forces
3. Second Law
    a. impulsive force
    b. constant force implies constant acceleration

4. Third Law
    a. for impulsive forces
    b. for continuous forces
5. Superposition Principle
    a. vector sum
    b. canceling forces
6. Kinds of Force
    a. Solid contact
        - passive
        - impulsive
        - friction opposes motion
    b. Fluid contact
        - air resistance
        - buoyant (air pressure)
    c. Gravitational
        - acceleration independent of weight
        - parabolic trajectory

During the design phase, we first sketch the scenarios and stories for the learning environment. When that was completed, the included critical concepts were adjusted and reinforced by refining the scenarios so that a large coverage of knowledge is possible.

6.5 Summary

This Chapter explains the design of the different agent's pedagogical function. Each individual agent possesses a set of unique heuristics under the principle of conceptual conflict. In addition, how the pedagogical agent detects and correct users' misconception is illustrated. Last but not least, it clarifies the motivation and details of the design of the learning tasks.

CHAPTER 7. SYSTEM FRAMEWORK AND ILLUSTRATION

This chapter first explains the system framework and discusses the system infrastructures. There follows two excerpts of the conversation protocols. The discussion will be closely referenced to the details of the scenarios.

7.1 System Framework

Our system has been implemented using the design framework of C-VISions, a socialized, collaborative, virtual interactive simulation learning environment. This framework is a generic, object-oriented software framework, and its design is based on the *Model-View-Controller (MVC)* architecture derived from the Smalltalk programming language. The *Model* component implements the virtual world and virtual objects. The *View* component implements the virtual world browser. It listens for events and renders them in the 3D browser. The *Controller* component implements support for actions taken by users in the virtual world. Figure 11 denotes the major system flow of our virtual environment.

The synchronization of events occurring in the virtual environment heavily relies on the C-VISions network component. This component keeps listening to the message from each client and propagates the decoded event across all the clients by means of broadcast network protocol.

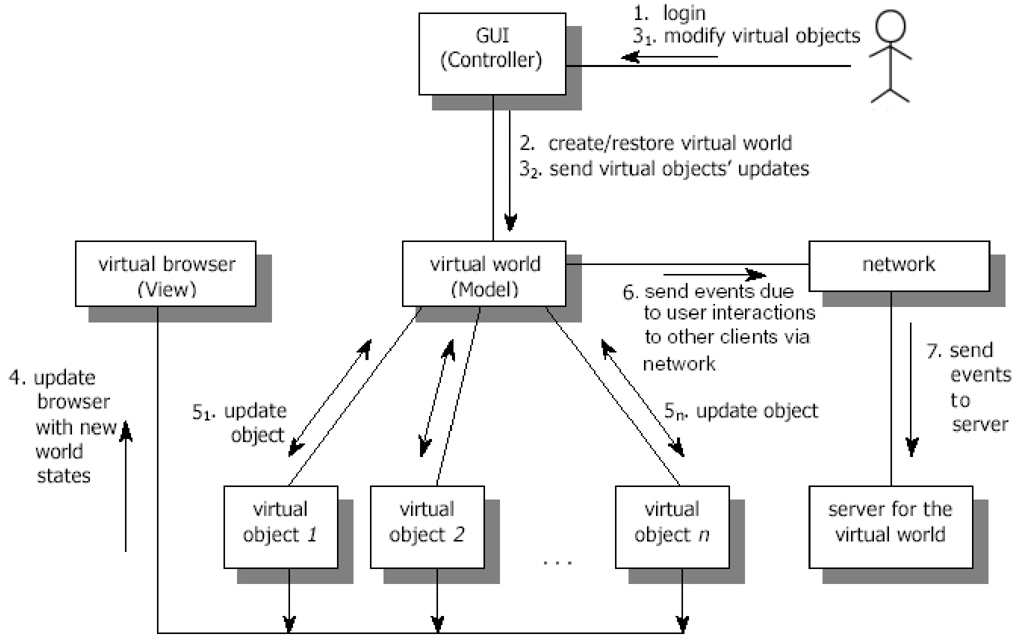


Figure 11. Schematic of flow control

To alleviate the workload of this C-VISions server, the communication between agents and users has been shifted on to a separate agent server. The arrows in Figure 11 indicate its information flow. Three agents reside together on the server machine where the agent communication takes place. They receive users' utterances or events from the clients over the TCP/IP network and send the decision back to the client after appropriate negotiation between the agents. The dotted ellipses on the client PC A in Figure 12 can be regarded as a virtual embodiment of agents and users as agent's decision and others' user manipulation are not made locally. This approach therefore makes use of a clear separation of agent's "mind" and "body", thus providing the convenience for agents to have a centralized control of their behaviors across every client PC.

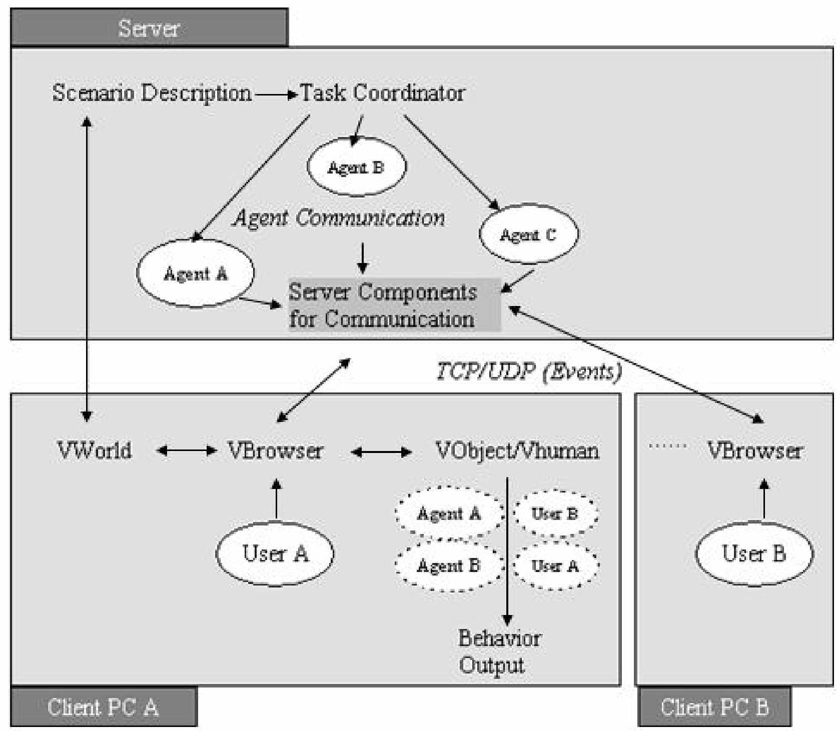


Figure 12. A Separate server to handle agent-user communication

The integration of our agent's architecture with the C-VISions framework can be further clarified with the aid of a layered structure diagram shown in Figure 13. At the bottom of the entire structure, the C-VISions system provides the requisite infrastructure for networking transmission and database administration. On top of that, the multimodal animation control and other output components, such as the text to speech engine, support the rendering of the agents or users' behaviors in the three dimensional virtual environment. Above this resides our intelligent agent architecture which enables the multiparty interaction. Task coordinator and scenario description sitting one level further up regulates the system flow and determines the goal of each agent. The top layer represents the domain knowledge as it imposes restrictions on the activities in the environment and the process of every below layer.

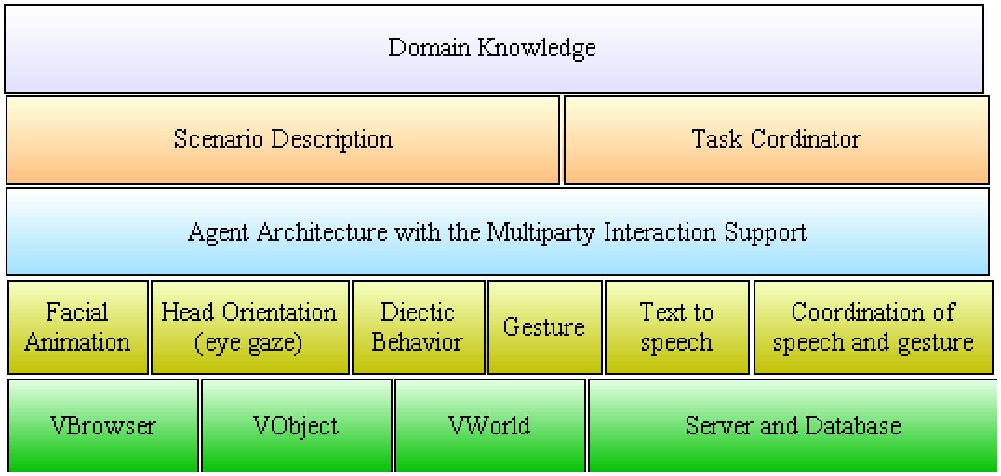


Figure 13. System integration with C-VISions

7.2 Environment Setting

A simulated virtual space station was constructed as the learning environment (see Figure 14). As mentioned before, there are three agents cohabiting the space station, namely, Ivan, Ella, and Tae. Ivan is assigned an instructor role to *tell* the correct procedures to the students; Ella contributes her expertise in helping students correcting misconception; Tae assists the students to overcome their thinking difficulties. Hovering in the sky, there is a spaceship which is one of the crucial object for the learning experiments. Some students get a chance to sit inside. On one side of the space station, students are able to see a square panel with blue lighting. The buttons on the panels enable students to either maneuver the spaceship or keep track of the behavior of any object within this virtual environment.

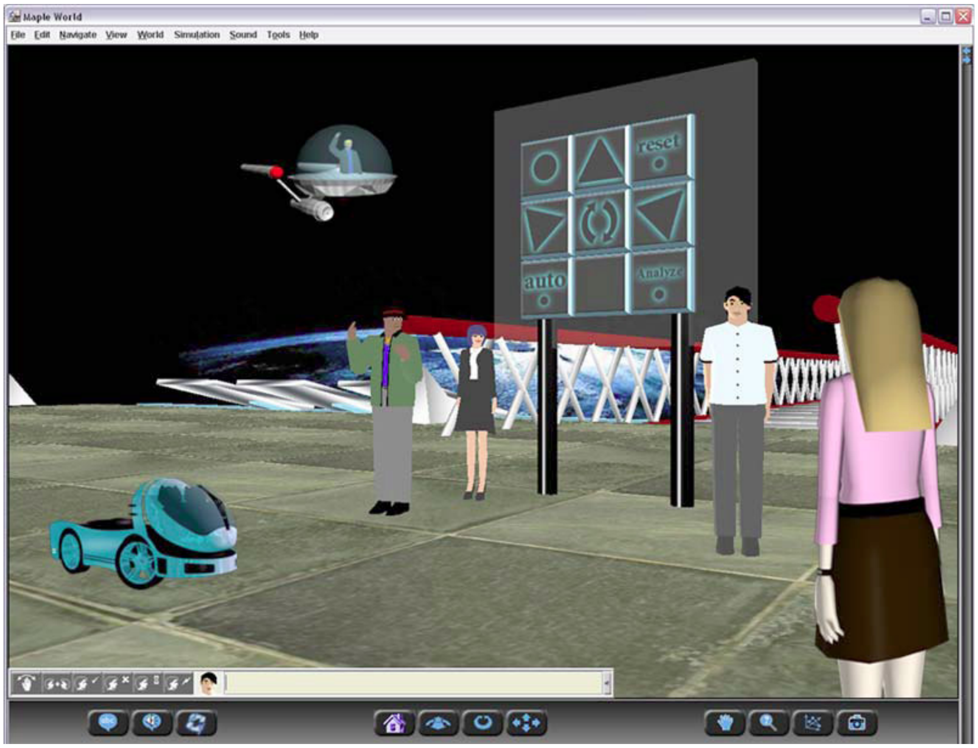


Figure 14. Users monitoring the moving vehicles from different perspectives

7.3 Illustrations

This section uses two prototypical episodes in the environment to demonstrate the different crucial aspects discussed in the previous chapters such as agent's layered architecture and multiparty interaction.

7.3.1 Agent Architecture

The next simulation protocol is a simple illustration of how the interaction among three agents and two users may proceed as part of the users' learning experience. The collaborations among different components of agent architecture are illustrated also.

The Protocol

		User Liu seems to be at a loss with the current task. He remains silent and does nothing.
1	Tae	Hi, Liu, What can I do to help you?
2	Liu	I really have no idea on proceed.
3	Tae	Could you be more specific about your problem?
		Liu did not reply. Tae calls Ivan for help.
8	Tae	Ivan, can you help Liu with the current task?
9	Ivan	OK.
		Ivan faces Liu.
10	Ivan	Liu, Let me reframe the task for you…
	. . .	
14	Ivan	Does my explanation make sense to you now?
15	Liu	Yah, I think so. I feel much clearer now.
16	Ivan	Which force do you think is larger; the force from spaceship to acceleration machine, or vice versa?
17	Liu	Possibly the first one.
18	Ella	Do You think the force from spaceship to acceleration machine is bigger?
19	Liu	Yes, I think so.
20	Ella	Could you let us know your reason?
21	Liu	I think it is bigger because the mass of the spaceship is larger.
22	Ella	Ivan, I'd like to suggest Liu do the gravity experiment
23	Ivan	Good idea. Please follow me, Liu.
		Another user Jay enters the virtual environment…
30	Tae	Welcome to our spaceship. …
		Jay walks towards Liu and Ivan. Noticing it, Ivan greets Jay.
36	Ivan	Hi, Jay. Come on, join us!
37	Jay	Yeah!
38	Ivan	We are talking about the relation between force and mass.
39	Ivan	Could you contribute some idea on this?
40	Jay	Hi, Liu, nice to meet you.
41	Liu	Nice to meet you too.
42	Jay	Ivan, how many tasks are there left?
43	Ivan	6
44	Liu	That is a lot. Let's hurry up.
45	Jay	Sure.
46	Liu	I don't think there is any relation between force and mass.
47	Jay	En… I am afraid I can't agree with you. . .
		…
53	Ivan	Can any one of you draw a conclusion?
		Et cetera . . .

Analysis

At line 1, after observing the idle behavior of the user, Tae's *behavior criticizer* identifies it as an indication of Liu's difficulty. Tae's *pedagogical model* is invoked, and the suggestions on improving the thinking process are provided. At line 9, the *intention capturer* in Ivan's understanding module recognizes that the utterance at line 8 from Tae implies both a verbal reply as well as a request for an action-to provide hints to Liu. At line16, Ivan asks a question regarding a common misconception encountered by previous users even though he knows Liu's answer is correct. At line17, a wrong answer is detected. At line 18, Ella issues a question to ascertain that the user did not respond spuriously. At line 21, a misconception is identified. Ella suggests that Liu performs a related experiment in order to correct the misconception. At line 30, Tae notices Jay's arrival. He knows that Ella has a higher priority to welcome new users. However, Tae's *perception system* detects that Ella is at present busy with another user; so Tae's *interaction controller* take over the role to welcome Jay. Just before line 36, Ivan's *interaction controller* is carrying out a "provide information" *interaction pattern*. Jay's arrival interrupts the *interaction pattern* so that Ivan's *interaction controller* pauses the "provide information" pattern, and instead a new "welcome" interaction pattern launches. When Jay has joined the conversation at line 37, Ivan's *interaction controller* resumes the previous interaction pattern to continue "provide information" and dynamically includes the user Jay as a new participator. He then invites Jay to share the ideas with Liu. Note that the discussion content (the relation between force and mass) is restricted by the Topic in the current task. From line 40 to 42, Ivan's *utterance analyzer* notices that the users do not follow the instruction to engage a learning discussion. However, Ivan does not immediately seek enforcement to restrict the users' freedom, but invoke *turn coordinator* to entertain the turn assignment by Jay in line 43. From line 46 to 52, Ivan's *dialog model* realizes that the group of user is undergoing a "discussion" behavior which is consistent with the interaction pattern. As a result, Ivan follows the instruction from his *interaction controller* to ask the users to conclude at line 53.

7.3.2 Multiparty Collaboration

The following scenario illustrates the multiparty collaboration when the agents make the efforts to help users improve their understanding of the concept *relative velocity*. Conceptual conflicts arise from time to time, giving rise to an interesting learning environment among users and agents.

Situation Narrative

In Figure 14, the students Mary (represented by the avatar wearing a pink blouse) and Jack (represented by the avatar wearing a men's grey suit) has the impression that the phenomenon of relative velocity only occurs in one-dimensional motion, because this is what what they have learned in class. This misconception is one of knowledge over-specialization. The evaluator agent, Ella (wearing a grey jacket over a white blouse), detects that both students share this misconception. Threfore, she requests

Ivan, the instructor agent (wearing a white shirt), to initiate a conflict resolution situation to extricate the students out of the misconception. Ivan invites Jack to teleport to a nearby spaceship and to observe the motion of a utility vehicle traveling along a straight path on the surface of the space station. The spaceship flies past at a low angle along a path parallel to the motion of the vehicle. Meanwhile, Mary also pays attention to the motion of the vehicle from the space station. Ivan then intentionally invites Mary to press one of the three directional arrows on the control panel to impose an instantaneous force on the spaceship, without Jack's knowledge. Mary presses the arrow in the left-most column of the second row of buttons. After the spaceship fly-past, Jack is teleported back to the space station. Ivan encourages Jack and Mary to share their observations with one another. Mary declares seeing the vehicle moving along a straight course toward her while Jack insists seeing the vehicle moving in a direction opposite to the spaceship's direction. Mary and Jack are able to reconcile their dissimilar observations by appealing to the concept of relative velocity applied in one dimension. However, Mary and Jack are unable to reconcile their mutual observations after Jack experienced the unexpected instantaneous force on the spaceship.

Figure 15. Bridging from percept to concept in the domain of relative velocity

To aid them in resolving this conflict, Tae, the conceptual thinking agent (wearing a green jacket) intervenes and invites Mary and Jack to compare videos of what they separately observed and to reflect on the differences. He then directs their attention to the screen on the right and asks Jack to guess which button Mary pressed while he was on the spaceship. (These buttons correspond to the *direction* arrows A, B, and C on the screen. Note that these arrows are not force vectors.) Jack conjectures the answer as the C direction, but Mary exclaims that she pressed the A direction arrow just now. Jack looks astonished, then confused. Tae, the thinking agent, asks Jack for the reason which makes him think direction C is the correct answer. Jack replies that it is because this is how things appeared to him as the spaceship moved toward the space station. Tae asks Mary to comment on Jack's explanation. Mary answers that it cannot be correct and proceeds to explain, with reference to the diagram on the screen, that direction C is actually the *resultant* direction that arises from combining the spaceship's initial velocity and the force applied in direction A. Ella nods approvingly at Mary, and Jack smiles weakly in apparent agreement. However, Jack continues arguing that, his observation indicates the car appeared to be moving *perpendicularly* toward him, with the side facing him; so he queries whether direction B should be the correct resultant direction instead. Tae asks Mary if she can resolve this dilemma for Jack. But Mary shakes her head slowly after pondering the request. At this point, Ella recognizes that Jack's observation of the car moving perpendicularly toward him is valid, and the spaceship moving in the resultant direction C is also valid. However, this was due to a very special situation: the amount of instantaneous force applied to the spaceship in direction A happened to reduce the composite velocity in the x axis of the spaceship to an amount equal to zero. In order to help the students recognize that this is a special case, Ella asks Ivan to set up another problem under a general case. Ivan then suggests that Jack and Mary re-perform the experiment. Ivan secretly increases the strength of the instantaneous force so that what Jack observes changes. This action leads to a fresh cycle of interaction between the students and the agents so that the students will not over-generalize from the results of the earlier special case. These cycles of interaction keep repeating until an equilibrium state of correct student conceptual understanding is achieved.

The Protocol

1		*The students, Mary and Jack, and the agents Ivan, Ella, and Tae, are gathered together on the space station.*
2	Jack	I am sure that the concept of relative velocity applies only to motion

		in one dimension. Yes; pretty sure.
3	Mary	Yeah; I think you are right.
4		*Ella detects that there is a problem of concept over-specialization; she asks Ivan to set up a conceptual conflict situation.*
5	Ella	Ivan, please come over and help these two students with the concept of relative velocity. [Rule 6]
6	Ivan	Oh, yes. Sure!
7	Ivan	Hey, Jack, I want you to observe the motion of this vehicle (*points to vehicle*) from the spaceship, OK? To teleport to the spaceship, press the top-left corner button on the control panel when you're ready. [Rule 2]
8	Jack	Roger.
9	Tae	Jack, what do you expect the motion of the vehicle to look like from the spaceship? [Rule 12]
10	Jack	Different from what is seen while standing on the space station, I guess.
11		*Jack teleports to the spaceship.*
12		*Ivan gets the vehicle moving along a straight path parallel to the path of the spaceship in motion. Jack and Mary observe the motion of the vehicle. Ivan records videos of what Jack and Mary see.*
13	Ivan	Mary, why don't you trigger a force on the spaceship? Give Jack a surprise, you know? (*Ivan grins.*)
14	Mary	Sure! Sounds like a great idea.
15		*Mary presses the first button in the second row of buttons. An instantaneous force is applied to the spaceship, altering its travel path. Mary and Jack continue observing the motion of the vehicle on the space station.*
16		*Ivan presses a button on the control panel to teleport Jack back onto the space station.*
17	Ivan	Jack, tell us what you observed. [Rule 4]
18	Jack	From the spaceship, the vehicle appeared to be moving backwards until, all of a sudden, . . .
19	Ivan	And Mary, what did you observe?
20	Mary	The vehicle moved at a constant speed towards me. . .
21	Jack	. . . boom! The spaceship jerked abruptly and started moving toward the space station.
22	Mary	. . . all the while.
23		Ha! Ha! That was because I applied an instantaneous force on the spaceship.
24	Tae	So do your observations agree with one another? [Rule 19]
25	Jack	Yes. Mary saw the vehicle moving forward but I saw it moving backward because the speed at which the spaceship was traveling was

		greater than the speed of the vehicle on the space station.
26	Mary	Yup; Jack is right.
27	Jack	But after the jerking force, I seemed to be moving toward the car.
28	Mary	That can't be! (*Mary seems momentarily confused.*)
29		*An awkward silence follows.*
30	Tae	Why don't you all have a look at the videos of what you saw earlier? Just go over to the movie screen on the left and play those videos. Then think carefully again about what you saw, OK? [Rule 13]
31	Jack	OK, Tae.
32	Mary	OK.
33		*Jack and Mary play through the videos on the move screen. Tae then directs their attention to the screen on the right.*
34	Tae	Jack, which button do you think Mary pressed? The one pointing in the direction A, B, or C?
35	Jack	Oh, I think it was direction C.
36	Mary	No way; I pressed the button pointing in direction A!
37		*Jack looks surprised.*
38	Tae	Jack, why did you think that Mary pressed the button pointing in direction C? [Rule 13]
39	Jack	Oh, this is how things appeared to me. The spaceship was moving toward the space station.
40	Tae	Mary, what do you think of Jack's explanation? [Rule 14]
41	Mary	It can't be right.
42		Direction A is actually the resultant of the spaceship's original direction and the direction C button that I pressed.
43		*Ella nods approvingly at Mary. Jack smiles weakly.*
44	Jack	But, from what I observed, the car was moving perpendicularly toward me.
45		Shouldn't direction B be the correct resultant direction instead?
46	Tae	Mary, can you help Jack to resolve this dilemma? [Rule 15]
47		*Mary thinks, then shakes her head slowly.*
48		*Ella recognizes that Jack's observation is indeed correct and Mary's answer is also correct.*
49	Ella	Ivan, can you help the students to understand that what they have observed is a very special case? [Rule 6]
50	Ivan	Ivan, can you help the students to understand that what they have observed is a very special case? [Rule 6]
51	Mary	OK.
52	Jack	Let's go!
		Et cetera . . .

Analysis

The rules indicated after certain sentences refer to the agents' heuristics mentioned in section 6.2. These heuristics endows the agents with the ability to lead effective interactions as well as promote collaboration.

In the beginning, the evaluator agent, Ella, detected the existence of concept over-specialization by noticing from the dialog history that both students explicitly affirmed that relative velocity operates only in one dimension. She also realized that the examples cited by the students were confined to one dimension only. So Ella requested the instructor agent, Ivan, to construct a conceptual conflict situation for the students to be involved in and to resolve. Ivan organized a group task for the students. They observed the motion of a common object, the utility vehicle, from two different frames of reference: the spaceship and the space station. But Ivan introduced an unexpected twist to the situation by asking Mary to impose a force to affect the spaceship behavior not anticipated by Jack. In doing so, Ivan engendered cognitive dissonance between the students.

In the meanwhile, he recorded the different views of Mary and Jack in the form of videos so that the conceptual thinking agent, Tae, can later make use of these videos to facilitate student reflection. Ivan sought to promote collaborative learning by asking Mary and Jack to share their mutual observations. Mary and Jack found mutual agreement in their understanding of what each saw before the moment when Mary imposed the instantaneous force. Nevertheless, they could not reconcile what each saw since that time.

The above impasse was detected by Tae. He intervened by asking the students to view the videos previously recorded. He did so to enable rebuilding of past observations, to highlight the contrasting observations, and to enhance cognitive dissonance. In doing so, the videos are deliberately juxtaposed on a common large screen.

To support the group thinking process, Tae asked Jack to identify the direction of the instantaneous force that Mary applied, first using the more abstract form of knowledge representation on the screen to the right. When Jack guessed that the force was applied in the C direction and Mary immediately contradicted him, Tae detected this obvious contradiction and proceeded to further scaffold the learning interaction between the students. To foster collaborative dialog, Tae asked Jack to state his justification for choosing direction C; he then asked Mary to comment on Jack's justification. Mary provided an informed response, providing evidence that she had some understanding of how velocity vectors combine to give a resultant velocity. She thereby earned the approval of Ella, as manifested by Ella's affirmatory nod. Jack appeared to be persuaded by Mary's explanation, but only barely. He quickly protests that, based on his observation, direction B is a possible alternative answer. Tae again tried to foster collaborative interaction, but Mary was unable to rebut Jack's suggested alternative. Ella recognized that Jack's observation and Mary's answer are both correct, but that this appears only in a very special case when the consequence of the force in direction A so happens to reduce the velocity of the spaceship in the horizontal direction to the same horizontal velocity of the utility vehicle. She detected that the students did not recognize the features of the present situation that make it a special case. Hence, she requested Ivan to set up a fresh learning experience that would help the students to experience a more general case that could be contrasted with the special case. In this way, the students' understanding can be deepened as they are made sensitive to the contexts of applicability of their knowledge.

7.4 Summary

This Chapter first gives details of the system framework of our learning environment. Various implementation issues such as system flow and server workload are addressed. After presenting the background of the learning environment setting, it discloses two scenarios to elucidate the underlying principles of the agent architecture.

CHAPTER 8. EVALUATION

The user empirical study described in this chapter sets out the evaluation objectives and the design of the evaluation methodology used. By analyzing the details of computer user interactions, we could not only evaluate the effectiveness of system and agent architecture but also collect users' learning behaviors for the input of progressive enhancement of agent's knowledge base.

8.1 Evaluation Objectives

This user study is to evaluate the *naturalism of the multiparty interaction* in the virtual learning environment developed. As highlighted in the first chapter, our research seeks to endow the agent with the ability to behave realistically as a member of a virtual group. The agent in our system has been incorporated with an interaction pattern control mechanism to manage an interactive situation involving many users and agents. Therefore, if the subjects describe that the interaction which occurred during testing is natural and effective, it is sufficient to demonstrate the effectiveness of the intelligent agent architecture we have developed.

The *naturalism of the multiparty interaction* in our evaluation is divided to two distinct levels:

Social level. If users subjectively feel that the agents are intelligent to generate believable social behaviors in a multiparty situation, it is likely an indication that our agent architecture endows the agent with the social intelligence to cultivate a natural multiparty interaction. Believable social behaviors here refer to the common social interaction protocol, such as greeting to the newcomer, and glancing at every listener during speaking. Successful achieving the social level naturalism of the multiparty interaction can enhance the mutual awareness of the agents and users.

Content level. If users subjectively feel the agents actively participates the interaction among multiple users and agents, such as comment on the content of interactions, or give suggestions after users have engaged in certain interaction mode such as discussion or debating, it is likely an indication that our agent architecture empowers agents with the capability to foster an effective group interaction. The content level naturalism of the agent often enriches users' experience and leads the interaction efficiently.

Learning effectiveness and efficiency is important but the secondary goal of this user study. We use pre and post test as well as evaluation sheet to analyze subjects' learning information. We believe that a detailed evaluation for learning effectiveness is more suitable after several iterations of user studies focusing on improving agent understanding of subjects' learning activities. It is because the continuous and sufficient feedback from subjects is necessary for us to revise agents' knowledge base to fit better under the current learning domain.

8.2 Methodology

We adopt a comparison approach to analyze the naturalism of interaction that users experience when using the system. To ensure a fair comparison, we have devised a simplified agent architecture which relinquishes the interaction pattern management, the dialog model and the function for agents to cooperate on the learning task from the agent architecture described in the section 3.2. However, the modified agent can still behave in a socially intelligent manner when interacting within a group. There are two underlying reasons for us to do so. First, since this simplified agent could still fulfill most of the believable social customs requirements recently proposed by the Mission Rehearsal Exercise project [16], it can be regarded as a representation of the current research status quo of handling multiparty interaction. Therefore, by comparing it with our agent architecture, the subjects could readily express the difference of their multiparty interaction experience with or without our enhanced agent's capability. This provides the opportunity for us to consolidate our conclusions about the benefits from utilizing our agent architecture. Second, as our system is the result of a multidisciplinary research, the overview of system without any emphasis perhaps makes subjects draw biased conclusion during evaluation. For instance, some subjects who are more interested in the facial and gesture animation instead of the multiparty interaction will draw inappropriate conclusion when evaluating the system. Our comparison setting allows us to focus merely on the agent's expertise of handling multiparty interaction.

8.3 Procedures

Six subjects participated in the user study. All of them experienced a virtual 3D environment before, either through computer games or some computer graphics courses. The user study was conducted individually. One of the researcher logged into the system together with every subject in order to simulate a three-agent two-user multiparty environment. Each user testing lasted approximately 45 minutes and there were six steps:

1. Read instructions: The users were clearly instructed about the goal of the study: to evaluate the *naturalism of the multiparty interaction*. In addition, they were briefed about the user testing process.

2. **Answer question in pretest:** There were two Newton's law related questions in the test. Although learning effectiveness is not the primary goal of the evaluation, we can collect useful findings when analyzing the testing result.

3. Interact in the multi-agent learning world with *limited* multiparty interaction support in agent's architecture: As described, the agents presented in this step possessed only the requisite social knowledge to generate behaviors in order to accommodate a multiparty situation. They would not actively involve subjects' group discussion or provide suggestions. However, other than that, there was no difference about the agent and environment setting when compared with the full functional agent

environment in the next step. Subjects were required to fill an evaluation form about their interaction experience immediately after finishing this step.

4. Interact in the multi-agent learning world with *full functional* multiparty interaction support in agent's architecture: The agents at this step utilizing our agent architecture were incorporated with the interaction pattern manager, the dialog model and other collaborative features for enriching the subjects' learning experience. Same as step three, subjects were required to fill an evaluation form to share their interaction experience at the end of this step.

5. Answer questions in the post test: The post test was exactly the same as the pretest. We would analyze the difference between pre and post test for understanding subjects' learning status.

6. Fill the evaluation form: The subjects were asked to complete the questions in an evaluation form.

The order of step three and four was reversed for three subjects out of the total six. This was to avoid the comparison biased due to the first impression of either system.

8.4 Observations

The observed results have been collected through observation, pre-test and post-test, questionnaire, and informal interview. These observations are grouped together based on the targeted evaluation dimensions defined previously.

8.4.1 Naturalism of Interaction

Tables 4 and 5 presents the results from the questionnaires related to the naturalism of the interaction for simplified agents (denoted as S Agents) and full functional agent (denoted as F Agents) learning environment respectively.

	1 - most strongly disagree 7 - most strongly agree	1	2	3	4	5	6	7	Avg/7
1.	I enjoyed interaction in such an agent assisted learning environment.		1		3	1	1		**4.16**
2.	I feel interested in engaging in the dialog with agents.			1	3	2			**4.16**
3.	The interaction is natural and rich as compared to the one in the real life.	1	2	1	1	1			**2**
4.	The agent could always take turn at an appropriate time during conversation.		1	2	2		1		**3.66**
5.	The agents have enough eye contact with me.				2	1	3		**5.16**
6.	The agents actively involve into the users' discussions.	2	2	2					**2**
7.	The agents actively involve into the dialog among other agents and me.	2	2	1	1				**2.16**

Table 4. Questionnaire result of naturalism of the multiparty interaction using simplified agent architecture

	1 most strongly disagree 7 most strongly agree	1	2	3	4	5	6	7	Avg/7
1.	I enjoyed interaction in such an agent assisted learning environment.	1		1	1	2	1		**4**
2.	I feel interested in engaging in the dialog with agents.			1		3	1	1	**5.16**
3.	The interaction is natural and rich as compared to the one in the real life.		1		1	2	2		**4.66**
4.	The agent could always take turn at an appropriate time during conversation.			2	1	2	1		**4.33**

5.	The agents have enough eye contact with me.				1	2	2	1	**5.5**
6.	The agents actively involve into the users' discussions.					2	3	1	**5.83**
7.	The agents actively involve into the dialog among other agents and me.			1		2	3		**5.16**

Table 5. Questionnaire result of the naturalism of the multiparty interaction using full functional agent architecture

All of the subjects were interested in interacting with one particular agent at the beginning. Two of them were attracted by Ella, and the rest were interested in listening to Ivan's instructions. Both the simplified agents (S agents) during step three and full functional agents (F agents) during step four actively joined the initial one-agent and one-user conversation. However, five subjects feel the conversation is more fluent for the F agents' environment because F agents always joined the dialog at a natural pausing point of the dialog, or a moment relevant to him/her. Below is an example.

```
[Caroline] Glad to meet you.
[Ivan] Glad to meet you, too.
[Caroline] Would you mind introducing your friends to me?
[Ivan] Sure, they are Ella and Tae.
[Ella] Hi, Ella here.
[Tae] Hi, I am Tae. How do you do?
```

Among all the six subjects, one claimed he was confused in figuring out the relationship and roles of the F agents at the beginning. Two subjects stated that they could not tell any difference of the interaction between F and S agent at the start. Nevertheless, four out of the six subjects felt the interaction with S agents become boring after the first 5 minutes. Although it is a multiparty environment, the activities that took place are more like “several parallel one-to-one interactions” instead of a realistic group interaction.

All of subjects felt that turn taking is managed well for any one-to-one dialog. This can be ascribed to the successful synchronization of the agent’s and the user’s speech voice, i.e. If a subject types the next utterance when the agent is speaking, the subject’s voice will only be heard after the speaker agent has finished. However, the F agents were described by one subject as occasionally “a little bit aggressive” during turn takings. It is because F agents are eager to get the turn to become the speaker whenever it thinks that is necessary moment according to the content of current group dialog. When more than one agent does so, the subject feels the dialog lacks appropriate pauses. Two subjects agreed they didn’t have sufficient time to comprehend the agents’ speech when the F agents continuously take over the turns one followed by another.

Four subjects were impressed by F agent’s ability to involve them in an ongoing dialog. F agents often could mention something related to the content of the dialog when joining a group discussion. The four subjects felt it very similar to the real world situation, in which the real human being adopts the same strategies to enter other’s conversation naturally.

8.4.2 The Effectiveness of Interaction

Table 6 presents the results from the questionnaires related to the effectiveness of the interaction for a full functional agent learning environment. The evaluation in this section emphasizes on how users feel the multiparty interaction atmosphere cultivated by agents benefits their tasks execution.

	1 - most strongly disagree 7- most strongly agree	1	2	3	4	5	6	7	Avg/7
1.	I found the agents are intelligent to handle a multiple users' situation. (e.g. give response according to a sequence of users' utterance.)			1		1	3	1	**5.5**
2.	Each agent's role can be clearly recognized.					1	2	3	**6.33**
3.	The cooperation among agents (e.g. agents will pass the user's query to the most appropriate agent to answer.) is effective and useful.			1		2	3		**5.16**
4	I will approach another agent if I realize the agent I intended to consult is busy talking with an other user.				2	2	1	1	**5.16**
5.	The cooperation among the agents helps me to understand the task better.		1		1	1	3		**4.83**
6	I feel it saves my time and energy when one agent actively responds if he has a better answer even he is not the intended listener.		1		2		3		**5**
7	Agents are helpful to suggest some activities for me to improve the understanding after s/he realizes my problem by analyzing my historical interactions with others.		1	1		2	1	1	**4.66**

Table 6. Questionnaire result of interaction effectiveness

All subjects could correctly associate an agent's name with its functional role after testing the system. They attributed it to the agent's clear role design as well as the consistent cooperation among agents which always enables the most appropriate agent to solve a user's problem. This result revealed our successful construction of agents' uniqueness by the complementary design of the agents' roles. Two subjects explicitly stated that it is very important to differentiate agents so that they will know who to approach whenever they hope to solve their difficulties in a short time.

Three subjects who first started interacting with the full functional agent world felt quite uncomfortable after the switch to the simplified agent world, since "the agent does not involve in users' discussion" any more. All of them considered it effective for F agents to provide suggestions during the users' conversation when necessary. One subject said "I was expecting the (S) agent to say something (during the discussion with another user), but he didn't."

Two subjects triggered only very few interaction patterns so that they did not receive too many agents' suggestions on their learning activities. The remaining four subjects showed their appreciation when agents provided the guidance they needed. When asked whether they could explicitly notice interaction pattern adopted by the agent, all of them declared they were not aware of it. As long as it does not impose too much restriction on their activities, all of the subjects felt that they enjoyed listening to the agent's advice because this makes them feel recognized in the virtual environment.

8.4.3 The Effectiveness of Learning

Table 7 includes the result of the user's feedback on learning effectiveness. Although learning aspect is not primary goal for this evaluation, we are able to identify some useful findings for the system refinement.

	1-most strongly disagree 7-most strongly agree	1	2	3	4	5	6	7	Avg/7
1.	The agents appear knowledgeable in the learning domain		1			3	2		**4.83**
2.	The task is explained and my questions are answered clearly.		1		1	1	3		**4.83**

3.	Agents could provide the learning assistant at the right time.	1		1	3		1		**3.66**
4.	Agents could identity and help me to recognize that I have physics misconception.	1		1	3	1			**3.5**
5.	Agents actively attempted to correct my misconception	1	1	1	2	1			**3.16**
6.	I feel that my understanding of Newton's physics has improved.		1		3	1	1		**4.16**
7	Through group discussion, I could better understand the learning context.			1	2	1	2		**4.66**

Table 7. Questionnaire result of learning effectiveness

Five subjects regarded agents as experts in the domain of Newton's law, because they delivered the tasks using clear structures and answer users' questions without many difficulties. However, the learning assistance generated in the real time such as prompting users for self-reflection, although innovative, was described as "primitive" by one subject. Almost all the subjects felt the suggestion provided by agents should be more complicated instead of only one verbal utterance.

Regarding the misconception detection and correction, only one subject was identified as possessing misconception. This subject later realized the problem himself after the evaluation agent tried to help him. Through speech act analysis, the intentions of all subjects were effectively determined during the preparation steps of the misconception detection. Nevertheless, the conversation history shows most subjects did not use the standard form to describe their understanding on the knowledge which made it difficult for the evaluation engine to transform those utterances into logical expressions.

We were pleased that five subjects concluded they could benefit from such an agent assisted multi-user learning environment. They felt the agent's multiparty interaction support have enhanced their learning collaboration experience.

Pretest and posttest comprise two questions related to relative velocity which is exactly consistent with the scenarios and tasks users have to undergo. The questions are listed below:

1. A boat is aimed directly across a river and its speedometer says 10 km/h. the captain of the boat knows that the current has a velocity of 4 km/h. What is the speed of the boat relative to the river bank? In what direction is the boat moving (relative to the bank)? What would be the boat's speed relative to the river bank if the current has a velocity of 10 km/h?"

2. An airplane's speedometer indicates that it is moving with a velocity of 120 m/s relative to the air. The compass indicates that the airplane is heading east. If the weather report says that there is a wind blowing toward the north at 30 m/s (relative to the Earth) at the plane's altitude, what is the airplane's velocity relative to the Earth? What would the plane's velocity be if the wind were blowing at 90 km/h toward the south?

Only one subject exhibited his difficulties on the knowledge of vector addition when solving the above question in the pretest. During the course of interaction in the virtual environment, he was reminded of the related concept and took part in a relevant experiment during the virtual interaction. He corrected his mistakes in the posttest.

8.5 Discussion

From the user study, we have a better understanding of our agent architecture when referring to the findings. They can be addressed into three categories: the interaction control, interaction dominance, and agent's understanding.

Interaction control

The user study has disclosed the enhancement of user experience by the approach of using interaction pattern. However, the improved multiparty interaction did not result in a satisfactory learning effectiveness. This can be ascribed to the inadequacy of learning focused interaction patterns. At the current stage, most of the interaction patterns defined are extracted from the real life common sense which lacks the support for learning effectiveness. Therefore, an additional research on extracting the interaction patterns is necessary.

Interaction dominance

Some subjects have raised the issue of dominance. They felt uncomfortable when three agents dominated the conversation most of the time. In a later informal interview, they insisted, in a conversation group consisting three agent and one user, it is not a good practice to allow each of them to share the 1/4 talking time. The user will feel agents are too aggressive. We realize it is a good idea to introduce the concept of agent and user dominance or activeness in the multiparty learning environment. It can be implemented into agent's profile and user's model, so that the most balanced combination of agents and users can be identified before the conversational group is formed.

Agent's understanding of user utterances

The user study tells us that an intelligent agent architecture entails robust interpreting ability. In our virtual environment, agent's understanding is realized through speech act classification, and the misconception identification additionally relies on an algorithm to transform a user's utterance to a logical expression. Refinement of speech act classification under our learning domain is necessary to enhance the accuracy of user intention interpretation. A fault tolerance mechanism for improving agent's recognition ability to convert user's utterance to logical expression is also essential.

8.6 Summary

This chapter first clarifies the evaluation objective: to analyze the naturalism of the multiparty interaction and followed by the procedures for user study. The evaluation questionnaires are presented with the explanation. Further discussion addresses the issues of interaction control, interaction dominance, and agent's understanding of user utterances.

CHAPTER 9 CONCLUSION

This chapter reviews different considerations when we implement the multiparty interaction support for the intelligent pedagogical agent. It also highlights the contributions and achievements of thesis. Last but not least, the prospect of the further work is discussed.

9.1 Research Summary

By considering multiparty interaction in the context of understanding, planning and teaching, our agents are designed to possess a high level of social and pedagogical intelligence in a multi-agent multi-user environment.

The agent's competency and capability of understanding was achieved by the adoption of an enhanced version of speech act classification: A dialog model tracks the users' conversations in both single-user and group-user modes hence permitting agents to interpret multiparty interaction. The considerations of the conversational roles enable the agent to identify the relationship among the multiple participates in a dialog which facilitates the process of intention capturing. The information of non-verbal user behaviors is utilized as an additional channel for agent to increase the accuracy of interpreting user's verbal utterances.

The efficiency of the task planning and discourse management in the agent architecture is accomplished through a multi-level topology. The components in this topology namely, *task topic*, *interaction function* and *interaction patterns,* not only decompose the learning task into small manageable aspects, but also effectively encourage the appropriate multiparty interaction styles which suit the learning purpose best. Both *interaction patterns* and *interaction functions* are invented to facilitate the multiparty learning activities.

Focusing on the pedagogical intelligence, each agent has been associated with a unique role. These roles complement with one another and maximize learning effectiveness and the user experience. A particular pedagogical ability of detecting and correcting misconception for Newtonian physics problem is developed also to improve students understanding during the course of the multiparty interaction.

A user study is conducted for evaluating the naturalism of the multiparty interaction in the virtual environment system we developed. A comparison approach is adopted. The analysis of the users' recorded interaction and their post evaluation feedbacks reveals the facts that our agent architecture can manage the multiparty learning interaction in a realistic and effective manner. Further discussions on how to improve the agent architecture raise attentions to the aspects of interaction pattern management, interaction dominance as well as agent's understanding on user utterances.

9.2 Contribution of the Thesis

The multidisciplinary work described in this thesis can be regarded as making an exciting beginning in multiparty environment research. The generic agent architecture that we developed can be easily integrated into other virtual environments under different domain. Users in these virtual environments thus can benefit from the efficiency engendered by the enriched interaction experience.

Enhancements to the existing speech act classification demonstrate the effectiveness of our methodologies adopted to interpret users' intentions with a combination of sources. It gives the inspiration for further improvement on natural language understanding for embodied conversational agents.

Although interaction patterns were not first introduced in this reveal, we have successfully implemented the idea in the context of our work for the first time. Additionally, the interaction model we have suggested is independent of the number of participating users which allows the agent architecture to achieve the greatest flexibility.

Identification of misconception was also part of the research in this thesis. The use of natural language as user's input undoubtedly creates big challenge for us. The misconception identification approach we have devised of applying a FOPC expression, although not perfect, addresses the problem creatively and generates the useful ideas of modeling student for future research.

Last but not least, the system we have deployed continues the C-VISions project with the idea of grouping multiple agents and multiple users. It not only elevates the research ambitions but also develops a practical system for students to learn Newtonian physics in an interesting manner.

9.3 Future Work

The agents with multiparty interaction support will become more and more favorable since there is a tendency that people will enjoy a "realistically crowded" virtual world, at least in a learning domain. This thesis emphasizes how the technology could enhance the learning interaction experience for students in the virtual environment.

Successful construction of the agent's knowledge base requires a complete understanding of students' learning behaviors. Therefore, further continuous user empirical studies are indispensable to the mature of the research. These studies will support the refinement of speech acts, interaction patterns, agents' heuristics, and misconception models, making it possible to improve agent's interpretation ability to a real human being comparable level under our specific learning domain. Additionally, the focus of the user evaluations should be gradually shifted from interaction naturalism to learning effectiveness.

In a technology aspect, a few directions worth exploring further. On one hand, the thesis does not elaborate much on the agent's multimodal animation when facing multiple parties. This actually could be an interesting topic to notably enhance the social believability of the agent. On the other hand, learning environment could scale up because interactions involving more than one group are the possible trend. Therefore, the management for both inter-group and intra-group interaction will raise new challenges to achieve efficiency and effectiveness in the tutoring process.

REFERENCES

[1] J. Cassell, T.Bickmore, L.Campbell, H.Vilhjalmsson, and H.Yan, "More than just a pretty face: conversational protocols and the affordances of embodiment," *Knowledge-Based System*, vol. 14, pp. 55-64, 2001.

[2] X. Yuan and Y. S. Chee, "Embodied tour guide in an interactive virtual art gallery," in *Proceedings of the International Conference on CyberWorlds 2003*. Singapore, 2003, pp. 432-437.

[3] J. Rickel and W. L. Johnson, "Steve: An Animated Pedagogical Agent for Procedural Training in Virtual Environments," in *Animated Interface Agents: Making Them Intelligent*, 1997, pp. 71-76.

[4] J. C. Lester, J. L. Voerman, S. G. Towns, and C. B. Callaway, "Cosmo: A Life-like Animated Pedagogical Agent with Deictic Believability," in *IJCAI97 Workshop on Animated Interface Agents: Making them Intelligent*. Nagoya, Japan, 1997, pp. 61-69.

[5] A. L. Baylor, "The impact of three pedagogical agent roles," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM Press, 2003, pp. 928-929.

[6] Y. S. Chee and C. M. Hooi, "C-VISions: Socialized learning through collaborative, virtual, interactive simultations," in *Proceedings of CSCL 2002: Conference on*

Computer Support for Collaborative Learning. Boulder, CO, USA, 2002, pp. 687-696.

[7] D. A. Kolb, "Experiential Learning: Experience as the Source of Learning and Development.," in *Englewood Cliffs*: Prentice-Hall, 1984.

[8] Y. S. Chee and Y. Liu, "Grounding concept in percept: Learning physics experientially in multi-user virtual worlds," in *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*. Joensuu, Finland: Los Alamitos, CA: IEEE Computer Society, 2004, pp. 340-344.

[9] J. R. Searle, *Speech Acts. An Essay in the Philosophy of Language*: Cambridge, 1969.

[10] A. C. Graesser, K. VanLehn, C. Rose, P. Jordan, and D. Harter, "Intelligent tutoring systems with conversational dialogue," *AI Magazine*, pp. 39-51, 2001.

[11] I. L. Seung and B. C. Sung, "An Intelligent Agent With Structured Pattern Matching For A Virtual Representative," in *Intelligent Agent Technology: Research and Development. World Scientific*, 2001, pp. 305-309.

[12] A. J.Dix, J. E.Finlay, G. D. Abowd, and R. Beale, "CSCW and social issues," in *Human-Computer Interaction*: Prentice Hall Europe, 1998, pp. 509-552.

[13] H. Sacks, "A simplest systematics for the organization of turn-taking for conversation," in *Language 50*, 1974, pp. 696-735.

[14] E. G. Padilha and J. Carletta, "A simulation of small group discussion," in *Proceedings of the sixth workshop on the semantics and pragmatics of dialogue (EDILOG 2002)*. Edinburgh, UK, 2002, pp. 117-124.

[15] E. G. Padilha and J. C. Carletta, "Nonverbal Behaviours Improving a Simulation of Small Group Discussion," in *Proceedings of the First International Nordic Symposium of Multi-modal Communication*. Copenhagen, Denmark, 2003.

[16] D. Traum and J. Rickel, "Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds," in *2nd International Conference on Autonomous Agents and Multiagent Systems*, vol. 2. Melbourne, 2002, pp. 766-773.

[17] F. P. M. Dignum and G. A. W. Vreeswijk, "Towards a testbed for multi-party dialogues," in *AAMAS 2003 international workshop on Agent Communication Languages and Conversation Policies*. Melbourne, 2003, pp. 63-71.

[18] H. J. Suh, "A Case Study on Identifying Group Interaction Patterns of Collaborative Knowledge Construction Process," presented at 9th International Conference on Computers in Education, 2001.

[19] E. Andre, J. uller, and T. Rist, "The PPP Persona: A Multipurpose Animated Presentation Agent," 1996.

[20] C. Elliott, J. Rickel, and J. Lester, "Lifelike Pedagogical Agents and Affective Computing: An Exploratory Synthesis," in *Artificial Intelligence Today*, M. Wooldridge and M. Veloso, Eds.: Springer-Verlag, 1999, pp. 195-212.

[21] J. Rickel and W. L. Johnson, "Integrating Pedagogical Capabilities in a Virtual Environment Agent," in *First International Conference on Autonomous Agents*. California, 1997, pp. 30-88.

[22] E. El-Sheikh and J. Sticklen, "A Framework for Developing Intelligent Tutoring Systems Incorporating Reusability," in *IEA/AIE*, vol. 1, 1998, pp. 558-567.

[23] F. Zelhart and E. Wallingford, "A Survey of Intelligent Tutoring Systems and the Methods Used for Effective Tutoring," 1994.

[24] J. Gregoire, L. Zettlemoyer, and J. Lester, "Detecting and Correcting Misconceptions with Lifelike Avatars in 3D Learning Environments," in *Proceeding of the Ninth World Conference on Artificial Intelligence in Education*. Le Mans, France, 1999, pp. 586-593.