Design and Implementation of a State Transition Model for Educational Robot Tutoring Math Homework

Shiyao Sang

Supervisor: Xinguo Yu

This thesis is submitted for the degree of

Master of Engineering

Central China Normal University Wollongong Joint Institute

Central China Normal University

September 2024

I would like to dedicate this thesis to my loving family and friends. I wish I could be a great magician in the field of Computer Science.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgments.

Shiyao Sang
September 2024

Abstract

Automatic problem solving has made significant progress in recent years, and its technology can already solve some of the simple math problems. Educational robots can be combined with automated problem-solving engines to tutor children's math homework and avoid the drawbacks associated with problem from Tipaipai model software. Because the math homework tutoring process is characterized by multiple rounds of interaction and multiple roles, the management of state transitions has its own complexity. The state transition management also directly determines the service quality of the system. The thesis first identifies the set of states in math homework tutoring, and designs a state transition model to cover the states. The thesis implements an educational robot tutoring math homework system based on the model, including a dialogue manager that completes multiple rounds of tutoring activity interaction and a multi-channel collaborative control system for the robot. Through experiments on top of the NAO robot, it is verified that the state transition of the model can effectively solve the problem of multiple rounds of interaction regarding homework tutoring and the problem of multiple roles participating in tutoring activities in homework tutoring.

Keywords: *educational robot, human-machine interface, finite state machine, dialogue system*

摘要

最近几年自动解题取得了显著进步，其技术已经可以解决一部分简单数学问题。教育机器人可以和自动解题引擎结合辅导儿童数学家庭作业，并避免题拍拍模式软件带来的弊端。因为数学作业辅导过程具有多轮次交互和多角色参与的特点，其状态转换管理也有其复杂性。而状态转换管理也直接决定了系统的服务质量。论文首先确认了数学家庭作业辅导中的状态集合，并设计状态转换模型涵盖了其中的状态。论文根据模型实现了教育机器人辅导数学家庭作业系统，包含完成多轮辅导活动交互的对话管理器以及机器人多通道协同控制系统。通过NAO机器人上面的实验，验证了模型的状态转换可以有效解决家庭辅导中的关于作业辅导的多轮次交互问题，以及多种角色参与辅导活动的问题。

关键词： 教育机器人, 人机交互, 有限状态机, 对话系统

Acknowledgments

After two years of study, I have finally graduated. I would like to firstly thank myself for my hard work in seeking and completing my studies, and for doing my best to learn what I wanted to learn. Secondly, I would like to thank my parents for supporting me in my studies. After all, there are many friends who had to give up their studies for financial reasons. Finally, I would like to thank my two supervisors and the people I met in the lab on my computer science journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background and Significance

Recently researchers from OpenAI and the Department of Mathematics at Peking University used a new method to improve the accuracy of automatic problem solving [1; 2], and OpenAI enabled the GPT-3 model to improve to a level of 55 points in answering elementary school math problems. In the automatic problem solving tournament held by the Chinese Computer Society in 2020, this answer was still correct at about 40%[3]. Although automatic problem solving has been studied since the 1960s, the overall accuracy results achieved remain low. Research in recent years has progressed significantly, and interpretable automatic problem-solving engines have been able to solve some simple math questions, and in the future, as technology advances, interpretable automatic problem-solving engines will gradually evolve into reliable and practical artificial intelligence functions.

To give full play to the role of automatic math problem-solving engines, we need to combine educational robots with them to form educational robot tutoring math homework applications. Based on this, educational robots will be able to assume the role of bridging the communication between children and teachers in the home education scenario and replace humans to explain the questions to children.

On the other hand, due to the development of the Internet and big data, Tipaipai and other photo search software have caused problems in school education. Children are using software to search for answers to questions instead of independent thinking. This is not good for improving children's problem-solving ability. In response to this situation, the Chinese education department issued a demand to take down the software of the photo-to-answer model [4]. In line with the previous policy carried out by the Chinese education department to try to reduce the repetitive schoolwork burden of students [5], The demand aims to reduce the length and quantity of homework and improve the quality of homework.

In the context of the double reduction problem, educational robots for tutoring homework in the home may be a good way to solve the legacy problems such as Tipaipai and protect students from the negation of AI, placing the application of AI in education under the control of teachers and experts. Educational robots will effectively avoid the negative effects of Tipaipai homework tutoring software, which can intelligently perform the task of homework tutoring for children, saving parents who lack teaching experience and are exhausted to some extent.

1.2 Current Status of Domestic and International Research

1.2.1 Current Status of Domestic Research

There are already many educational robots for home tutoring in the market. These educational robots generally tend to be homogeneous in terms of function, with the function of tutoring students' after-school homework and expanding homework-related knowledge. However, these educational robots only have the ability for single-round interaction and lack the ability for multi-round interaction. To some extent, they are more like a storyteller or a video player. As shown in Fig. 1.1, the educational robot product Alpha Egg [6] from iFLYTEK has the

Fig. 1.1 Alpha Egg [6]

functions of storytelling, reading textbooks aloud, and knowledge quizzes. But it is more like a video player with a robotic appearance that can express expressions. The main interaction method of Alpha Egg is to interact with children through voice commands and auxiliary control screens. It does not support multi-round interaction functions.

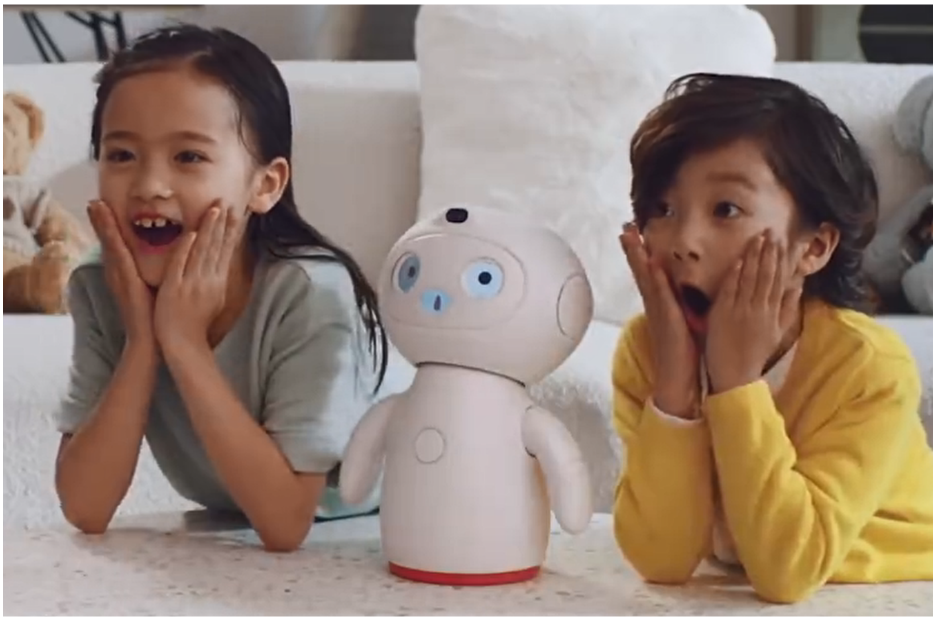


Fig. 1.2 Xiaoyi Elf[7]

Huawei Whole House Intelligence and Smart Screen Flagship New Product Launch Event in 2021 released the children's educational companion robot Xiaoyi Elf [7], as shown in Fig. 1.2. The Xiaoyi Elf has the function of expression display and physical expression, and an intelligent Q&A function. The video in the launch shows the Xiaoyi Elf can perform a variety of educational games with children. But the video was labelled as an artistically processed video. And the Xiaoyi Elf don't enter the sales process after the Launch Event. Other videos about the Xiaoyi Elf showcase only show the Xiaoyi Elf with a single question-and-answer function. Although Huawei's children's educational companion robot can finish the exploration of collaborative puzzle games with children, it cannot interact with multiple rounds of homework tutoring scenarios.

After analyzing the functions of educational robots on the market for homework tutoring, we can find that these educational robot products are only able to complete the knowledge explanation ability under single-round interaction. But tutoring homework is a multi-round interaction process. These robots lack the ability to perform a long time and multiple rounds of homework tutoring.

In China, some researchers have already started to explore the problem of multiple rounds of interaction in educational scenarios. Researchers from Neusoft Research Institute investigated the application of multi-round dialogue technology in knowledge quizzes in online learning [8]. They apply multi-round conversational chatbots to solve the problem of repeated questions and answers about knowledge points in teaching.

1.2.2 Current Status of International Research

International research on the application of educational robots has been conducted for many years. Related researchers conduct a wide range of research from early childhood education to higher education. Educational robots are used by researchers for teaching a second language [9], learning mathematics [10], learning programming [11], and other related functional applications.

Notable studies in this regard are those on educational robots regarding collaborative learning. LaVonda Brown and Ayanna M. Howard attempted to use educational robots to increase social interactions to encourage students and improve the student's math scores in math exams [12]. Researchers in Israel used robots to assist students in higher education to manage the learning process to improve their learning [13]. Huili Chena investigated three different roles of educational robots as teacher, companion, and student of children. It is believed that educational robots that can switch roles adaptively can improve student achievement [14]. These studies have shown that robots can interact beneficially with students to assist their learning and improve their study.

After multi-round technologies have matured and been applied in commercial scenarios, multi-round technologies have been gradually tried to be applied in educational scenarios. Shu Matsuura, a Japanese educational researcher, used multi-round dialogue technology to assist teachers in extending the question in the lecture[15]. Shu Matsuura used a multi-round chatbot to visually demonstrate the meaning of entities in the teaching process. on the other hand, Shu Matsuura uses a robot to naturally introduce the question to the experiment by conversing with the robot. The human-like robot multi-round dialogue approach in the teaching not only complements the information of the teacher's lecture through the robot's words but also brings a friendly atmosphere to the classroom.

In addition to this, multi-talk technology has also been applied to teaching and learning oral language [16; 17]. These are a large number of conversational scenarios in oral quizzes. JinXia Huang used a role-play dialogue system for language learning [18]. He breaks down scenarios into subtasks with a certain degree of freedom in order. The dialogue system also supports non-thematic dialogue to increase the flexibility of the dialogue. Sherry Ruan in Stanford University applied multi-round dialogue technology to the reading of digital materials [19]. Children engage in a multi-stage dialogue with a chatbot through four stages–

Introduction, Recommendation, Learning, and Assessment. Different stages of the dialogue have different degrees of freedom. The system can assist children to read digital materials.

1.3 Existing Problems

After analyzing the current state of domestic and international research, research on educational robots tutoring math family activities need be researched. In the educational robot math home tutoring process, the educational robot needs to assist students in the ongoing process of math problem solving. This is a complex ongoing multi-turn interaction process, and in order to achieve this process, the robot needs to complete many state transitions.

On the other hand, robots tutoring math homework may be faced with multiple roles involved in home tutoring. This requires the educational robot to have the capability to tutor multiple students at the same time and to tutor students accompanied by parents. In addition, teachers should be able to intervene in the tutoring process of the educational robot so as to guide the educational robot's homework tutoring, as well as to indirectly relate to children and participate in the robot's tutoring of homework. This requires the state transition management of the state transition model of the educational robot to be able to accommodate multiple roles involved in homework tutoring.

After analyzing the current state of domestic and international research, research on educational robots math home tutoring activities needs to be studied. In the educational robot math home tutoring process, the educational robot needs to assist students in the process of math problem-solving activity. This is a complex multi-round interaction process. To finish this process, the robot needs to complete related state transitions.

On the other hand, robots tutoring math homework may be faced with many situations. Multiple roles may involve home tutoring. This requires the educational robot to have the capability to tutor multiple students at the same time and to tutor students accompanied by parents. In addition, teachers should be able to intervene in the tutoring process of the educational robot to guide the educational robot's homework tutoring, as well as to indirectly relate to children and participate in the robot's tutoring of homework. This situation requires the state transition management of the state transition model to be able to adapt multiple roles involved in homework tutoring.

Problem 1: State transitions in educational robots tutoring math homework need to be studied.

Problem 2: The management of state transitions for multi-role participation in educational robot tutoring math homework activities needs to be studied. That is situations where the robot tutors multiple students at the same time, where the parent-student participates in tutoring together, and where the teacher directs the robot homework tutoring system.

1.4 Main Research Content of the Thesis

This thesis hypothesizes that in the scenario of educational robot tutors math homework, the children are 8–12 years old and have independent thinking and learning abilities. The children can accept a long interaction with the robot without adverse effects on their mental development. The educational robot is controlled by both teachers and experts. Teachers are responsible for guiding the robot's tutoring activities. Experts are responsible for the research of the educational robots and the analysis of student interaction behavior.

In this thesis, the NAO robot is used as the platform for the study. To address the problems in educational robot homework tutoring, the thesis designs a state transition model for educational robot tutoring math homework, which is used to solve the problem of multiple rounds of interaction in robot tutoring math homework. The state transition model determines the quality of robot services. The model also considers the rationality of the state transition model in the case of multiple entities involved in homework tutoring activities. After testing the educational robot tutoring math homework system implemented by the model, it is verified that the model can perform the multi-round interaction task of robot tutoring math problem-solving homework, while the model works well in the case of multiple actors involved in robot tutoring homework activities. Those are the case of robot tutoring multiple students at the same time, the case of robot tutoring students accompanied by parents, and the case of the teacher guiding the educational robot's tutoring activities. The main contributions of the thesis are as follows.

1. The thesis designs a state transition model for educational robots tutoring homework.

In this thesis, the tutoring task state of the robot tutoring math homework is obtained from the math problem-solving process model. After fully considering the states needed for system implementation and the states required by individual students, the set of states–Question, Answer, Explain, Check, and Wait–is determined for the educational robot in homework tutoring. The thesis designs a state transition model that covers all states in the set of states and gives ethical constraints that need to be taken into account when the model is implemented. The designed state transition model also considers the rationality of state management in the case of multiple roles involved in the home tutoring activities. The state transition model can be adapted to the case where the educational robot tutors multiple students, the case where the educational robot tutors students accompanied by parents, and the case where the teacher instructs the robot to tutor homework activities.

2. The thesis implements a multi-round dialogue manager for homework tutoring activities based on the model.

The dialogue manager can implement the state transitions in the model during the tutoring process and complete tasks in the home tutoring activities. The multi-round conversation manager is led by a hybrid of user and system, with the student's voice commands driving the tutoring process and the system obtaining the missing entities from the student through discourse. The multi-round dialogue manager uses a Hierarchical finite state machine as the control method. The basic state of the model is used as the state of the first layer of the finite state machine, the details of the model implementation and the student's thought process for solving the problem, and the excessive state of the student's model transitions in the system as the second layer of the finite state machine. The model uses a user state manager to store the states of each different student and parent, and each student establishes a separate flag and its profile so that the educational robot can distinguish multiple student interactions. Thus the model can adapt to the situation where the educational robot tutor multiple students at the same time. The dialogue system considers the parent as a special interaction object, and the parent can participate in the explanation of the educational robot's tutoring activities and help the student understand the robot's explanation in a more student-friendly way. The parent controls the educational robot to jump to the next state from Explain State. Teachers can indirectly participate in the educational robot's tutoring process for families by configuring a library of questions generated by the educational robot. The teacher-configured question dataset is a collection of questions that contains a table of contents of multiple questions, question tips, and answers to the questions and explanations of the questions. Teachers can configure the questions manually. In the future, they can select questions based on a recommendation system and a future automated problem-solving engine. Teachers can view the robot's interactive log records between student and robot to observe actual homework completion and make timely adjustments to the student's instructional plan.

3. The thesis summarizes two design approaches for the collaborative control of multi-channel outputs in educational robot home tutoring activities.

In a robotic environment lacking a hardware abstraction layer, educational robot implementations can label the content of action sequences, LED states, and voice responses for multi-channel outputs through an XML intermediate language. By parsing the action sequences recorded in XML, the educational robot implements the multichannel collaborative output in the home tutoring activities. In a robot environment with a hardware abstraction layer, the educational robot can pre-design the action sequence group, which includes robot action sequences and robot LED changes. The educational robot accepts the command parameters passed by the dialogue manager and outputs the reply contents in a language synthesis, and uses an action label parser to parse the action sequences therein and invoke the action stored in the hardware abstraction layer. The second approach was chosen for the implementation of the thesis. The benefit of this implementation is the ability to combine the output of complex actions through a visual action editor as well as existing simple actions. Action design and parsing costs are reduced compared to XML label parsing which can use the action interfaces already available in the hardware abstraction layer.

1.5 Outline of the Structure of the Thesis

The thesis has six chapters in total, and the content and structure of the thesis chapters are as follows.

(1) Chapter 1 is the introduction of this thesis, which introduces the research background and significance of educational robot homework tutoring, lists the current status of research at home and abroad, and the existing problems in the current status of research. Then, according to the existing problems in the study, the main research questions of the thesis are described. finally, the structure outline of the thesis is introduced.

(2) Chapter 2 presents the relevant theories and techniques of this thesis. It mainly introduces the theories and related techniques used in the thesis, including the two classical mathematical solution process models that the model relies on, the behaviourist learning theory, the natural language understanding techniques and the multi-round dialogue system techniques required for the model implementation, the finite state machine and its extended hierarchical finite state machine that implement the state control of the multi-round dialogue system, and chain-of-responsibility design pattern to handle the request and reply message in the dialogue system

(3) Chapter 3 is the design of the state transition model for the educational robot tutoring homework in this thesis. Firstly, two classical mathematical problem-solving process description models are introduced to determine the requirement when students solve the math problem. On the basis of these requirements, the tasks of tutoring activities and the corresponding robot states that the educational robot needs to undertake in the process of mathematical problem solving are derived. Based on the full consideration of the tutoring task state requirements in the home tutoring activities, the state required for system function implementation, and the students' personal state requirements, the thesis summarizes the set of states of the robot in homework tutoring—Question, Answer, Explain, Check, and Wait. The thesis designs a state transition model that covers all the state sets. The state transition model also can be effectively adapted to a variety of situations in homework tutoring. Those are the case where the educational robot tutors multiple students, the case

where the educational robot can adapt to students accompanied by parents in the model, and the case where the educational robot can receive guidance and intervention from teachers.

(4) Chapter 4 is the system implementation of the state transition model in this thesis. The main content is the educational robot tutoring math homework system implemented according to the state transition model of educational robots in homework. The system includes a cloud service layer, a conversation manager layer, and a robot layer. The cloud service layer consists of a problem database, an interaction log database, and a teacher management tool. The conversation manager layer is implemented by a conversation manager, which contains a natural language understanding module, a finite state machine module, and a user state manager module. The robot layer consists of the robot's action module, LED module, and sound module, which form a multi-channel collaborative control system.

(5) Chapter 5 is the experimental part of this thesis. The main content is the experiment of the verification of the model state by the educational robot tutoring math homework system. Through the experiments on the educational robot homework tutoring system, it is verified that the model can accomplish the task requirements in the homework tutoring process. Through experiments on multiple roles involved in educational robot tutoring activities, it is verified that the model can effectively adapt to multiple roles involved in educational robot tutoring activities. Thus, it is demonstrated that the educational robot tutoring home state transition model design can not only cover the multi-round interaction requirements of tutoring math homework activities, but also meet the model transition requirements of multiple roles participating in tutoring activities.

(6) Chapter 6 is the conclusion and future work section of this thesis. It summarizes the research contents of the previous five chapters and concludes the thesis with future research directions for the model of educational robot tutoring homework.

Chapter 2

Related Theories and Techniques

2.1 Problem Solving Model

Problem-solving is an important element of teaching and learning mathematics. Problem-solving helps students to think mathematically. The problem-solving model is a process model about the internal or external description of the problem-solving process. The internal description refers to the metacognitive process in problem-solving, and the external description is a staged description of the problem-solving process. In this regard, Pólya is one of the most prominent researchers in this area.Pólya proposed the famous 4-step problem solving model. Pólya divided problem solving into four steps: (i) understanding the problem, (ii) devising a plan, (iii) carrying out the plan, and (iv) looking back [20]. He believes that by following these four steps, one can discover successful solutions to problems. The cognitive load can be reduced by this generalized strategy, which is a heuristic approach to problem-solving.

For the study of the math problem-solving process, other researchers have followed its line of research and the extension of its study. Many of these studies have explored models of problem-solving exploration and experimentation, with Yimmer and Ellerton's study focusing more on the description of the problem-solving process from the perspective of preservice teachers: (i) Engagement, (ii) Transformation, (iii) Implementation, (iv) Evaluation, and (v) Internalization [21].

Yimer and Ellerton's model compared to Pólya's model divides the last stage into two parts, Evaluation and Internalization. the last stage of Pólya's model emphasizes checking answers and finding alternative solutions, as well as thinking about the methods used. Yimer and Ellerton's model distinguishes them. The Evaluation stage checks the results, and the Internalization stage reflects on the knowledge learned from problem-solving and the expectation of applying it to future problems.

2.2 Pedagogical Learning Theories

In the development of learning theories, three major learning theories have been developed that have been influential in the instructional design-behaviourist theory of education, cognitivist theory of education, and constructivist theory of education.

Behaviourist learning theories were developed from behavioural psychology. Behaviourism emphasizes the stimulus-response connection, and behaviourist learning theories are also called stimulus-response learning theories. Behaviorist Skinner believes that learning is a process in which an organism acquires new experiences by forming contextual-behavioural associations as a result of reinforcement of the outcome of a response in a situation. Behaviourist learning theory emphasizes the importance of repetition and review. They advocate the use of external rewards and punishments to control the learning process. Educators should arrange the environment so that learners can respond appropriately to stimuli in order to reinforce appropriate behaviour to the greatest extent possible.

Cognitivist learning theory was developed from cognitive psychology. Cognitivist learning theory focuses on consciousness and emphasizes understanding and thinking within the individual. Cognitivist learning theory views learning as the formation and reorganization of the learner's internal mental structures and emphasizes the connection and interaction between the logical structure of the learning content and the learner's existing cognitive structures. Concretely expressed in real teaching activities, educators should focus on the learners' perception of themselves and their learning environment to make knowledge meaningful. At the same time, educators should provide appropriate problem situations based on learners' existing mental structures and guide learners to use what they have learned to solve new problems.

Constructivist learning theory aims at the active construction of knowledge. Constructivism is "student-centred" and aims to move from the traditional focus on external management to self-directed, self-regulated learning. Constructivism hopes that instructional design will stimulate the individual learner's motivation and subjectivity so that a deep and broad summary and connection of knowledge and experience can be constructed within the scholar, rather than a simple, superficial understanding, to form a truly universal instructional experience that can be widely transferred and applied elsewhere.

2.3 Natural Language Understanding Technology

Natural language understanding is a field of natural language processing in artificial intelligence that attempts to allow computers to understand or extract information from natural language. The main task of natural language understanding is to transform unstructured text into structured text. Natural language understanding contains sub-tasks such as text classification, named entity recognition, relation extraction, and reading comprehension. Intent recognition is also known as intent classification. Intent classification belongs to the problem of text classification. Intent recognition can assign the text output by the user to a classification of intentions in a predefined domain. Intent recognition helps the computer to determine the intent of the text entered by the user and understand the operational tendency of the meaning that the user wants to express. Named entity recognition refers to the recognition of entities with specific meanings in the text, mainly including names of people, places, institutions, proper nouns, etc., as well as words such as time, quantity, currency, and proportional values. The process of named entity recognition generates named entities with proper noun annotation information from unstructured text. Slot filling is an application of command entity recognition for dialogue systems. Through named entity recognition, the dialogue system can determine whether the current question contains and can extract the required information and use it for subsequent dialogue. Slot information is usually used as parameter information for intent recognition and constitutes the basic function of natural language understanding in dialogue systems.

2.4 Dialogue System Technology

Functionally, dialogue systems are divided into question-and-answer systems, task-based dialogue, and open domain chat. The question and answer system focus on the question and answer of relevant knowledge, the construction of the knowledge base and the function of information retrieval. Users can complete information acquisition through dialogue with the Q&A system. Task-based chatbots are mainly domain-oriented multi-round dialogue systems. Task-oriented chatbots usually acquire the user's intention and related entity information through intention recognition and named entity recognition, and carry out dialogues on task contents in the domain. And the entire conversation system is controlled by a conversation manager. Open-domain chat does not limit the content of the chat, and there is no explicit task or subject of the chat. In the development domain chat, users can ask any question they want, but it is difficult to construct more than two rounds of conversation in the development domain. Development domain chat usually exists as a casual chat when the task-based conversation cannot find the corresponding task subject. Development domain chat can be enriched with chat content.

The human-computer conversation process can be divided into three types: user-driven, system-driven, and hybrid-driven. The user-driven process is where the user leads the operation of the entire dialogue system. The system runs with the user's questions and commands and responds accordingly, such as a question and answer system. System-driven dialogue systems usually have the system initiate the dialogue and the user responds to the system's dialogue, such as a restaurant robot actively asking the user for ordering information. In a hybrid-driven dialogue system, both the user and the system can take ownership of the dialogue. The user and the system can alternate the dominance and obtain the information that both parties want to obtain. For example, in a multi-round human-machine dialogue, the user sends a request to the dialogue system. The dialogue system does not obtain the slot information in the request uploaded by the user and needs to verify with the user the entities required for the user-related slot information for subsequent dialogue needs. This process occurs a transition from user-driven to system-driven. So the dialogue system in this process is a hybrid-driven human-computer dialogue system.

The content of the dialogue system framework is divided into five main parts. Automatic Speech Synthesis, Natural Language Understanding, Dialogue Manager, Natural Language Generation, and Speech Synthesis. The knowledge base is the source of data on which the dialogue system relies in the human-computer interaction process. As Fig .2.1 shows the dialogue system framework.

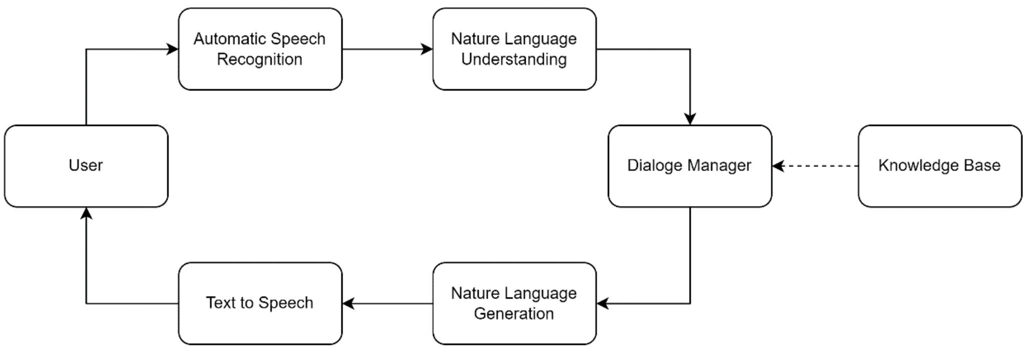


Fig. 2.1 Dialogue System Framework

The main functions of the Conversation Manager are conversation state tracking and conversation management policies. Dialogue state tracking can maintain the current dialogue state based on the semantic representation accumulated throughout the dialogue history. The main function of the conversation manager's management policy is to advance the continuous interaction of human-computer conversations through certain control policies. The dialogue manager is divided into three main approaches.

The first approach is a rule- and template-based dialogue system. The early applications of this area were mainly simple chatbots that could perform simple reasoning based on templates. Later, methods such as finite state machines and slot-filling methods emerged.

Finite state machines divide the task into a finite number of states and control the state transfer of the dialogue system by satisfying certain conditions. The finite state machine approach is simple and reliable, but finite state machines rely too much on expert experience, and the transfer path grows exponentially as the number of conversation states increases. The slot-filling approach views the interaction between the system and the user as the collection of information from the user by the system, and will continuously replenish the information in the slots that need to be collected regardless of the order of the user input until all slots are identified when the corresponding responses will be returned. Nowadays, the mainstream task-based dialogue system frameworks are based on the combination of finite state machine approach and slot-filling method, which takes the slot filling situation as the condition for state transfer, such as API.AI, WIT.AI and other dialogue system frameworks. However, this combined approach is often limited by the accuracy of the speech recognition module and natural language understanding module received by the system.

The second approach is a data-driven approach based on the use of Marker processes and Bayesian networks to select strategies for dialogues. This area of research is represented by a 2005 study from the University of Cambridge, UK[22]. Since the accuracy of speech recognition in the past was difficult to fulfill the accuracy requirements for information input in the dialogue system, the dialogue system had difficulty in determining the current state. In each round of interaction, the dialogue system inferred whether the current message was valid or not and performed three states: keep, delete, and ask. And rewards are given. This data-driven approach inferred the best response by maximizing the reward over the past conversation history. This data-driven approach can get rid of the tediousness of manually designed state transfer paths. However, this approach is rarely applied due to the large amount of manually labeled data required and the difficulty of constructing a limited user action space and machine action space.

The third approach is a deep learning-based dialogue system, where each module of the dialogue system is replaced using a neural network or built uniformly from input to response using an end-to-end deep learning model. A typical deep learning-based model encodes the input text into sentence vectors via a seq2seq model and decodes the words of each reply based on the sentence vectors. The advantage of this neural network model over data-driven models is that it requires less manual annotation and solves the problem of larger input and output space. However, this neural network-based model still requires a large amount of data, and it is difficult to solve the security response problem, as well as the neural network model may have a universal meaningless answer. In general, neural network-based dialogue systems are often used as a corpus supplement to rule-based dialogue systems or to optimize the modules contained in rule-based dialogue systems, such as Microsoft XiaoIce [23], which uses neural network models as the second layer in a three-layer response system.

2.5 Finite State Machines and Hierarchical Finite State Machines

A finite state machine is a mathematical model of computation. It is an abstract machine that happens to be in one of a finite number of states at any given time. A finite state machine changes from one state to another in response to some input; the change from one state to another is called a transition. A finite state machine consists of an initial state, a list of states, an initial state and a definition of the inputs that trigger each transition.

A hierarchical finite state machine is a type of finite state machine. Layering is a design useful in software design, requirements analysis and software testing. A hierarchical state machine divides the system into separate states and also places the states as sub-states in the current state. These sub-states can also be finite state machines. Hierarchical state machines not only solve the problem of modularity and reusability by introducing parent-child state machines. It also helps those involved understand more complex systems by using a hierarchy of components.

2.6 Chain-Of-Responsibility Design Pattern

Design patterns are normative prescriptions for software model design summarized from past software engineering development and represent best practices in software engineering. Design patterns are generic solutions to general problems faced by software developers in the software development process. These solutions are descriptions or templates for solving problems that have been summarized by many software developers over a considerable period of time through trial and error. There are 23 design patterns [24], which can be divided into three categories: Creational Patterns, Structural Patterns, and Behavioral Patterns.

The Chain of Responsibility design pattern [25] is a behavioural pattern. Chain-of-responsibility design patterns create a chain that handles the sender's messages. This pattern decouples the senders from the receivers. Each receiver has an application to the next receiver. If the current receiver cannot process the current sender's request, the sender's request is given to the next receiver and so on, until the message is processed.

2.7 Summary

This chapter focuses on the relevant theories and techniques used in the thesis. The chapter begins with an introduction to the two mathematical problem-solving models on which the state transition model is based and the behaviorist theory of educational learning embodied in the problem-solving model. Then the dialogue system techniques required for the model implementation and the concept of hierarchical finite state machines used in it are presented. Finally, the message reply mechanism implemented within the system based on the chain-of-responsibility design pattern is presented.

Chapter 3

Educational Robot Tutoring Math Homework State Transition Model

3.1 Variable States in Problem Solving Models

Problem-solving is an important part of teaching and learning mathematics. Mathematics homework is a form of training in problem-solving in mathematics learning. Research on problem-solving aims to characterize the internally described, externally described processes in problem-solving. Following these steps helps people to discover solutions to problems and reduces the cognitive load. In this regard, the famous scholar Pólya proposed the famous 4-step problem-solving model Pólya divides problem-solving into four steps: (i) understanding the problem, (ii) devising a plan, (iii) carrying out the plan, and (iv) looking back[20]. Yimmer and Ellerton extended his research from the perspective of pre-service teachers by proposing a five-step problem solving model: (i) Engagement, (ii) Transformation, (iii) Implementation, (iv) Evaluation, and (v) Internalization [21].

In Yimer and Ellerton's five-step problem-solving model, Transformation and Implementation are the process of thinking and implementing the students' solutions to the problem. Internalization is the digestion and absorption of knowledge related to the question. For math homework tutoring by educational robots, Engagement, Evaluation, and Internalization are the stages in which the tutor can participate in the student's math problem-solving process.

In the Engagement phase, the robot can select appropriate questions for the student, allowing the student to receive more appropriate training for his or her situation. Depending on the student's situation, the robot can also help the student to understand the problem by parsing some of its implicit conditions. So the Question state should be a demanded state in the design of the homework state of the educational robot tutoring.

In the Evaluation phase, the educational robot can check the answers to the questions for the students. According to behaviourist education theory, the question training conducted for homework requires timely feedback and knowledge reinforcement. But the reality is often that if students are given answers, it is difficult to ensure that they will not copy the answers and it is hard to explain answers in time. But with the use of robot tutoring for homework, students can effectively achieve answer checking during the completion of their math homework. So Answer state should be a state demanded in the design of educational robot tutoring homework state.

In the Internalization phase, the educational robot can explain the problem analysis to students. In the Internalization phase of the five-step problem-solving model, students need to organize their own thinking about the question and the knowledge behind the question. The educational robot can help students complete the Internalization phase of the problem-solving model by explaining the analysis of the question. So the Explain state is one of the required states in the educational robot tutoring homework state design.

3.2 Educational Robot State Transition Model

Based on the analysis of the problem-solving model, we summarize the three required task states for the problem-solving process in which the robot tutors students to complete their math homework: question, answer, and explanation. However, for the complete operation of the robot system, other states are needed to assist. For a single educational robot tutoring homework with students, the interaction state transition can be summarized using several states of question-answer-explanation, but math homework often requires multiple question composition. In order to achieve a cyclic tutoring state transition for multiple questions, the robot system needs a state to determine whether the student has completed the math homework. This gives rise to the "Check" state. The Check state is used to check if the student has completed all the math homework. If the student completes all the homework, the student will be given praise. If the student does not complete the homework, the student will move to the next tutoring state. So the original educational robot tutoring math homework states transition becomes a cycle of Question–Answer–Explain–Check states.

Since students have more than just problem solving related states in real life, students may need to stop tutoring homework for something or students may need to take a break in the middle of solving a problem. So this requires the robot system to enter a "Wait" state. After the student resumes the problem-solving state here, the robot system should be able to return to the previous state.

So the educational robot state transition model requires five states—Question, Answer, Explain, Check, and Wait—to complete the task of tutoring homework. The Question-Answer-Explain-Check cycle completes each round of tutoring homework state transitions. Waiting is a special state in the tutoring process where the robot system enters a waiting state for students to handle special matters and necessary breaks in the problem-solving process. Since the question may not be suitable for the student, the student can request to skip or change the question, and the robot system can jump back to the Question state from the Answer state. Also since the Answer state and Explain state is a long time state, the student jumps to the Wait state. In addition, if the student does not understand the interactive output of the educational robot in the tutorial state, the student can ask the educational robot to re-execute the robot tutoring math homework activity in the current tutorial state again. The state transition model of the educational robot is shown in Fig. 3.1. Table. 3.1 shows the tasks of the educational robot tutoring homework in each state.

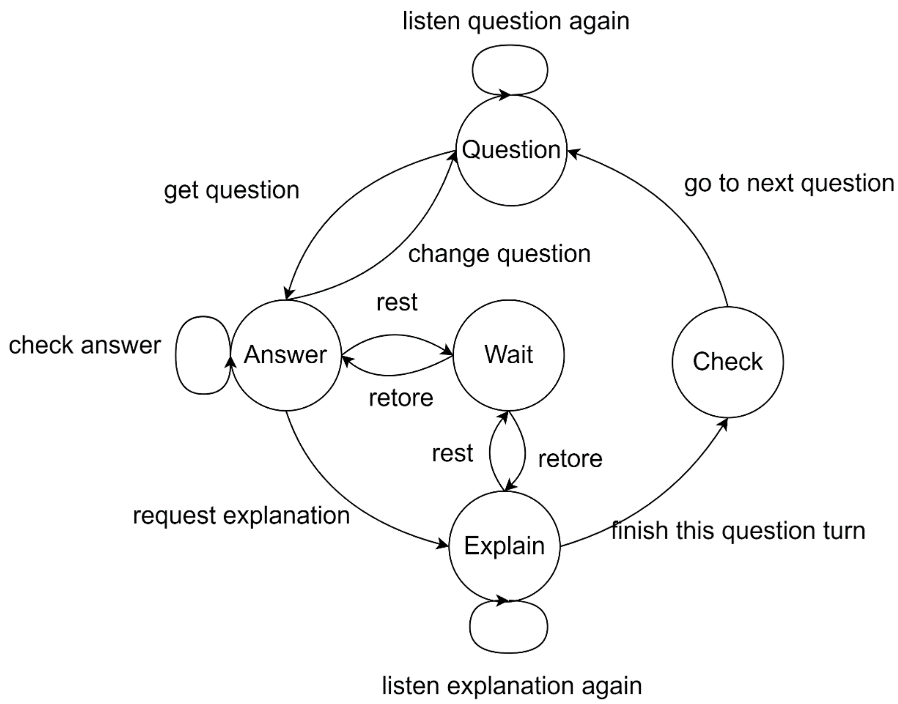


Fig. 3.1 Educational Robot State Transition Model

3.3 Sub-State Transition Model for Important States

Since the task state of the problem solving model is still a complex process. In the following, the important tutoring states in the educational robot tutoring math homework system will be

Table 3.1 Tasks of Educational Robots in Tutoring Math Homework States

Question
1. The robot home tutoring system select questions that are appropriate for the students ability and use them to reinforce lecture knowledge. The selected questions will be composed of a set of questions. 2. The robot select a question from the set and tell the students the direction of the question. 3. The robot give students tips to help student understand the question according to their abilities.
Answer
1. Students complete the solving of the question. 2. The robot tell students hints to solve the problem and guide them to solve the idea. 3. The robot tell the students the answers to the questions and check the questions. 4. After checking the answers, students rethink the questions and answer them again if they are wrong.
Explain
1. The robot explains the problem to the students. 2. Students can change different source of explanation. 3. Students think about the question again and finish absorbing the knowledge of the question. 4. Students answer the questions again according to the explanation.
Check
1. The robot home tutoring system checks if the student has completed the question in the question focus, and if not, starts the next round of the problem-solving process. If they do, the student is praised. 2. The robot home tutoring system records the student's interaction in the log database, which is used for student learning analysis and system improvement.
Wait
1. The child has a emergency to handle and he needs to interrupt the problem-solving process. 2. The child needs to take a break because he/she is tired of solving the problem. 3. The robot home tutoring system can save the student's current state and restore it when the Wait state is over.

further analyzed. For Question, Answer, and Explain states should be decomposed into more sub-states to be implemented, and how they implement the tutoring tasks in the problem solving model will be discussed.

3.3.1 Sub-State Transition Model for the Question State

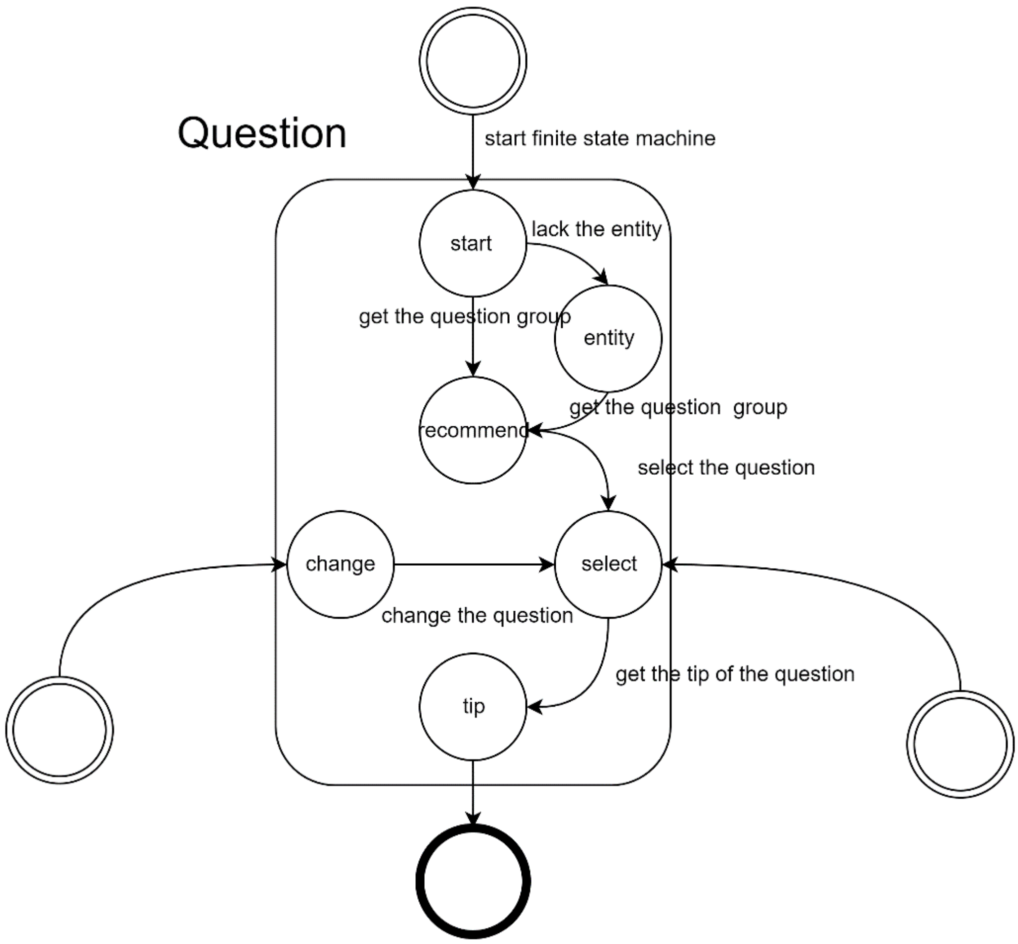


Fig. 3.2 Sub-State Transition Model for Question State

The Question state is responsible for generating the set of questions, selecting the initial questions, and the question comprehension hints. The Start and Entity states are responsible for getting the set of questions selected by the student. Recommend state is responsible for reading the set of questions from the database based on the selected set of questions obtained. Select state is responsible for giving the initialized questions, typically the first question of the question, and telling the student the question list information. Then the question hint is read and the student is told the question hint. This process completes the selection of questions and the use of question hints to help students understand the questions.

3.3.2 Sub-State Transition Model for the Answer State

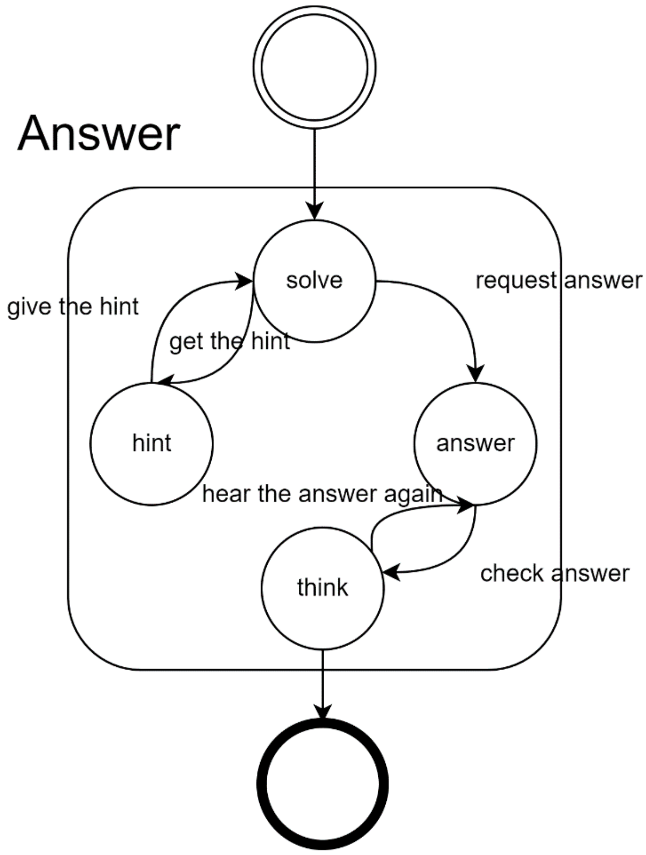


Fig. 3.3 Sub-State Transition Model for the Answer State

The Answer state is responsible for maintaining the student's solution state. In this state the Answer state should help and guide the student to think about the question and answer it, as well as to check the answer, and in the Solve state the robot waits for the student to solve the question. During this process, the student can ask the robot to give hints about the answer. After the student finishes solving the problem, the robot will give the answer checking function in the Answer state. After checking the answer, if the answer is correct, the student can think about the question further in the Think state; if the answer is wrong, the student can re-solve the question. At the same time, the student can repeat the request for the unheard answer.

3.3.3 Sub-State Transition Model for the Explain State

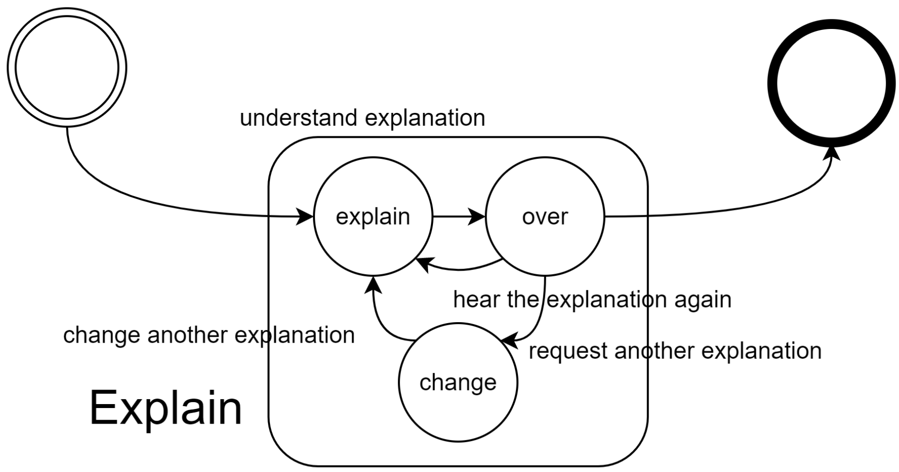


Fig. 3.4 Sub-State Transition Model for the Explain State

The Explain state is responsible for helping the student to understand the question. After the student finishes the Answer state, the robot explains the question in the Explain state to help the child understand the question. After the explanation, the robot will wait for the student to think in the Over state. If the student does not understand the explanation, he or she can ask the robot to repeat the explanation. Or if the student wants to change to another question, he/she can request another question by changing the state.

3.4 State Transition Model For More Situations

The state transition model for educational robots is built for situations where educational robots tutor a single student in question-answering learning, but in real home tutoring, there are often multiple roles involved in the home tutoring activities. This requires a management strategy for the state transition model that can accommodate multiple roles.

3.4.1 Teacher Participation Situation

Homework is an extension of schooling, and teachers want to be able to guide the homework tutoring process. This requires an indirect association between the educational robot tutoring home state and the teacher. The teacher configures the question database manually or using a recommendation system and an automatic problem-solving engine. Teachers can configure matching math homework for the current course or for knowledge points. Teachers select homework questions for the Question phase to form question groups and the necessary question solving hints to participate in the Question state of the home load state transition model. The teacher can add both answers and explanations when configuring homework. This way the teacher also participates in the Answer and Explain state parts of the homework tutoring state transition. The questions added by the teacher will determine the operation of the Checker state. So in the case of teacher participation, the teacher can actually influence the four states of the model indirectly or directly, while the last Wait state is for the student's own rest and temporary other business considerations.

3.4.2 Multiple Students Participation Situation

Families are likely to have multiple students who are not at the same grade and ability level. This requires the model to be able to accommodate situations where multiple students are involved at the same time. The model can be seen to consist of a single round of tutoring on a question and a Wait state. The model is a tutorial on a single question for a single student. The set of questions is iteratively cycled through the state transition model to complete the tutoring process for all questions, and in fact, there are multiple state transition models that are executed sequentially in time. Similarly, for the case of multiple students, we can build multiple state transition models simultaneously to handle tutoring of multiple students in the family. Each student will be initialized with a number of different question sets that drive multiple parallel home tutoring state transition models, thus processing the tutoring process for the students in the home on multiple question sets in parallel. The interaction between each student and the robot will have a label about the student, which drives the corresponding homework tutor-like transformation model for that student. Thus, the robot's state transition model in homework tutoring can be implemented to perform homework tutoring tasks with multiple students simultaneously.

3.4.3 Parent Participation Situation

In the home, parents accompany students to complete homework tutoring. Parental involvement in student learning can also improve student achievement [26], [27]. Because of individual differences, the explanations of educational robots may not be suitable for every student to understand. As parents, they are more mature adults to understand the explanation and are more familiar with students' communication and expression habits. The parents can listen and then assist students in understanding the educational robot's explanations. So the educational robot tutoring homework state transition model needs to adapt to the situation where parents and students are mixed. The first design is that the parent-student talk is treated as noise completely and does not activate the state transition of the robot system. In the other design, the parent can drive the state transition model directly from the explanation state to the checking state after explaining the problem to the student once more. This design avoids interrupting the student who is thinking about the parsing and allows the student to drive the model from the explanation state to the inspection state. The latter design will allow parents to participate in the educational robot tutoring math homework state transition model and will make it easier to control the state transition of the model while the parent accompanies the student to complete the homework.

3.5 Summary

This chapter focuses on the state transition model for educational robot tutoring math homework. The thesis first introduces the set of states required for educational robot tutoring math homework and designs a state transition model covering all states. Then the state transitions between sub-states under important states are discussed in detail according to the state transition model. Finally, the state transition model is discussed as a management strategy for solving the problem of multi-role participation in the homework tutoring process.

Chapter 4

Educational Robot Tutoring Math Homework System

4.1 System Framework

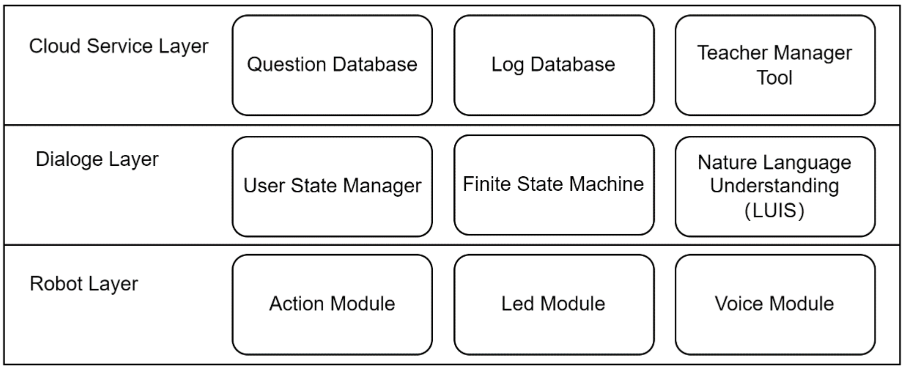


Fig. 4.1 Educational Robot Tutoring Math Homework System Framework

The educational robot tutoring math homework system is used to implement a state transition model for educational robot tutoring homework and to provide math homework tutoring directly to students. The educational robot tutoring math homework system includes three layers. Cloud service layer, dialogue layer, and robot layer. The cloud service layer includes a question database, log database, and teacher management tool. The question database contains the question group number, question information, question answer, and question explanation. The question information contains the table of contents for the questions and the questions. The log database contains the state transitions in the math problem-solving assignments of the educational robot tutor. Teacher management tools include tools for manual configuration of the question database. The future will include a question recommendation system and an automatic problem-solving engine. The question recommendation system completes the recommendation of questions, and the automatic problem-solving system generates answers and the content of explanations. The dialogue manager consists of a natural language understanding module, a finite state machine, and a user state manager. The natural language understanding module processes the user input passed by the robot and provides intention classification and named entity extraction to the finite state machine module for jumping. The finite state machine is a direct implementation of the state transition model for educational robot tutoring homework. The educational robot tutoring homework state constitutes the first layer of the finite state machine, the basic state of the model as the first layer of the finite state machine, the details of the model implementation and the student problem-solving thinking process, and the student's model transition over the state in the system as the second layer of the finite state machine. The user state manager records each student's state and identification information for the parents, which is used to identify each different object of interaction. Each object can share the structure of the finite state machine and maintain its own current state. The robot layer consists of an action module, an LED module, and a sound module. The robot action module is responsible for controlling the robot's action system, the LED module is responsible for controlling the robot's lighting system, and the voice module is responsible for controlling the robot's speech synthesis and speech recognition, as well as the voice recognition used to confirm the user's identification information.

4.2 Cloud Service Support System

4.2.1 Question Database

The question database records the data of math questions used by the robot for tutoring math homework. The question database is actually a knowledge base of questions required by the dialogue system. The question database is entered manually by the teacher or the person concerned, and in the future, the questions are generated by the recommendation system, and the answers and explanations are automatically populated using the automatic problem-solving engine. The design of the question database is shown in Table. 4.1.

4.2.2 Log Database

The log database records the state transitions that occur during the robot system's interaction with students. With the observation of the log database, teachers can observe students'

Table 4.1 Question Table

Name		Question Table			
Name of table		QuestionTable			
Field Name	Field Description	Field Code	Data Type	Primary Key	Not NULL
Id	table index	Id	int	yes	yes
Question	the index and tips of question	Question	nvarchar (50)	no	yes
Answer	the answer of question	Answer	nvarchar (50)	no	yes
Explanation	the explanation of question	Explanation	nvarchar(50)	no	yes
Note	the note of the question	Note	nvarchar(50)	no	no
Class	the group of the question	Class	nvarchar(20)	no	yes

Table 4.2 Log Table

Name		Log Table			
Name of table		LogTable			
Field Name	Field Description	Field Code	Type	Primary Key	Not NULL
Id	table index	Id	int	yes	yes
User	the name of user	User	nvarchar(20)	no	yes
State	the state of user in this interface	State	nvarchar(20)	no	yes
Time	the time of interface	Time	datatime	no	yes

homework learning situations. Also, the maintenance designer of the robot system can improve the robot system based on the log system. The design of the log database is shown in Table. 4.2.

4.2.3 Teacher Management Tools

The teacher management tool is an aid for teachers to manage the question database. The teacher management tool is used to add and manage the question database. Teachers can add question data manually. In the future, teachers can use the recommendation system to automatically generate relevant questions and use the automatic question-solving engine to generate answers and explanations. Fig. 4.2 shows the teacher adding questions page. Fig.

4.3 The interface of the question form after the question database after the teacher adds the question. The teacher can also view the log database to evaluate the students' learning effect and adjust the teaching plan in time based on the state transition and the time record of the transition when the robot system tutors students' homework.

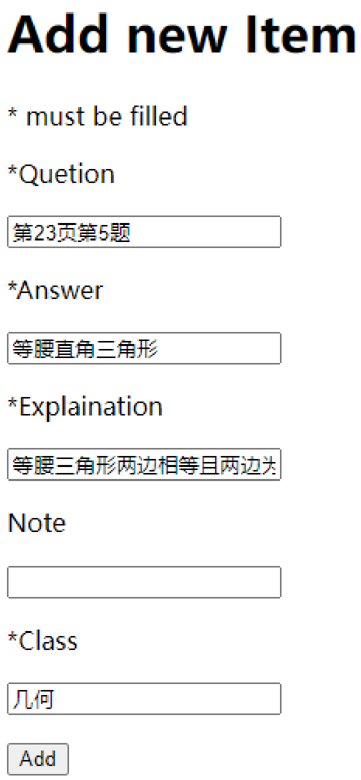
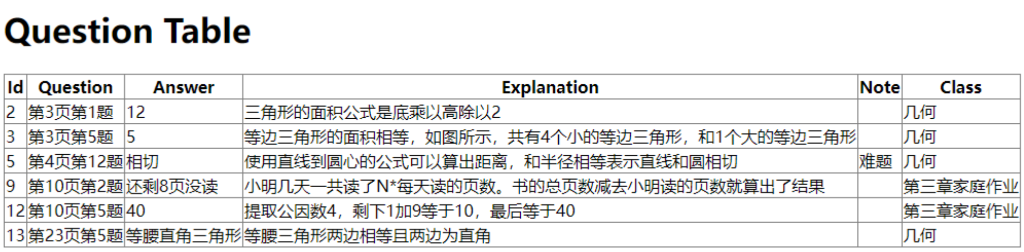


Fig. 4.2 Teachers Add New Questions Page



Question Table

Id	Question	Answer	Explanation	Note	Class
2	第3页第1题	12	三角形的面积公式是底乘以高除以2		几何
3	第3页第5题	5	等边三角形的面积相等，如图所示，共有4个小的等边三角形，和1个大的等边三角形		几何
5	第4页第12题	相切	使用直线到圆心的公式可以算出距离，和半径相等表示直线和圆相切	难题	几何
9	第10页第2题	还剩8页没读	小明几天一共读了N*每天读的页数。书的总页数减去小明读的页数就算出了结果		第三章家庭作业
12	第10页第5题	40	提取公因数4，剩下1加9等于10，最后等于40		第三章家庭作业
13	第23页第5题	等腰直角三角形	等腰三角形两边相等且两边为直角		几何

Fig. 4.3 Teachers Questions Management Page

4.3 Dialogue Manager

The Dialogue Manager is a direct application of the state transition model for educational robots tutoring math homework. This dialogue system is primarily used to verify the implementation of the major states in the state transitions and to complete a typical specific math homework tutoring process. This system will implement most of the state transitions in the state transition model used to support the tutoring process, with some of the state transition details omitted due to the ease of system implementation.

4.3.1 Dialogue Dominant Strategies

The dominant strategy of a dialogue system determines the dominance of the dialogue system and the user for the current dialogue initiation. Dialogue domination strategies are classified as user-dominated, system-dominated, and hybrid-dominated. In this paper, we adopt the hybrid domination approach. On the one hand, the student needs to change the state of the dialogue manager by voice command driver, on the other hand, in the topic state, the robot math homework system, the robot needs to name the entity to identify the group name of the math problem in the process of the student requesting the problem group, which requires the robot to be able to verify the specific math problem group name to the student. So the dominant strategy of the dialogue is mixed dominant.

4.3.2 Dialogue Input Control Strategy

The dialogue input control strategy is responsible for the processing of the input to the dialogue system. The dialogue control strategy accepts the processing results from the speech module and provides the processing results to the dialogue state control module. The usual processing flow of dialogue input is from automatic semantic recognition to natural language processing and finally to the dialogue control strategy. However, since the task of multi-role participation in family counseling activities needs to be solved in this paper, it is necessary to use voice recognition as well as user manager to differentiate the corresponding users; this function also provides the verification of legitimate users. The specific processing flow is as follows.

The student's voice commands are uploaded to the multi-round dialogue manager via the robot's voice recognition module, including voice text information for automatic speech recognition and user identification for voiceprint recognition. First, the user state manager in the dialogue manager will verify the user's identity based on the voice-recognized user identifier. Secondly, the natural language understanding process handles the voice text

Table 4.3 User Status Manager Configuration

Name	Description	role in the system
user	User Name	Identification of the current user
group	question group name	used to confirm whether the user's classification is student or parent
questionIndex	question index	used to confirm
state	current state of the robot system	used to save the current state of the robot system
stateback	state stack	for storing the robot state before jumping to the waiting state
questionlist	question list container	used to store the list of questions of the current question group read from the question database

information to find the results of the intent recognition will be evaluated and the corresponding named entity recognition. Finally, the natural language understanding module will give the processing results to the dialogue state control strategy. As shown in Fig. 4.4 the dialogue input control strategy for robot tutoring math homework.

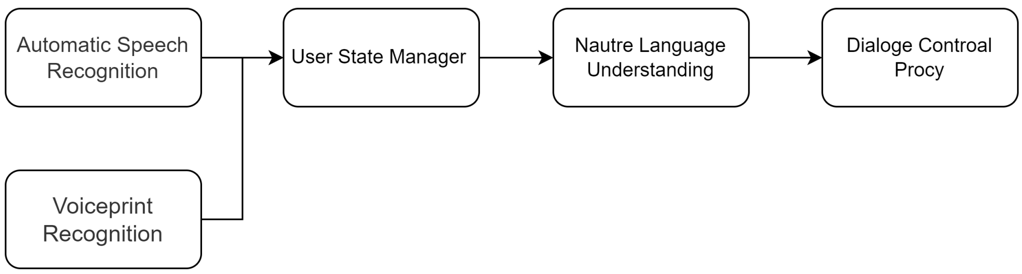


Fig. 4.4 Dialogue Input Control Strategy

The user state manager verifies that the user identification result of the input voice recognition is the currently interacting user. If this user exists in the query configuration information, the voice text message uploaded by this user is processed for natural language understanding. If this user is not entered as the current user in the manager, then the entered voice text message is treated as noise and the voice request for this upload is discarded. The user state manager configuration information is designed as shown in Table. 4.3.

The natural language understanding module performs semantic understanding processing. Natural language understanding will perform intent recognition and named entity recognition on the text. If the confidence level of intent recognition is lower than the set value, it means that the uploaded voice information does not belong to the existing intent and the voice command will be discarded. If the intent recognition is higher than the set value, it will indicate that the uploaded information belongs to the existing intent, and the text recognized by this speech command and the recognized naming body will be discarded. In this thesis, Microsoft LUIS Semantic Understanding Service is chosen for natural language understanding processing. Compared with other semantic understanding services in China, it has better recognition accuracy. Educational robot tutoring homework system with 8 intentions. Fig. 4.5 shows the intent settings in Microsoft LUIS, and Fig. 4.6 shows the intent templates in the start_question intent.

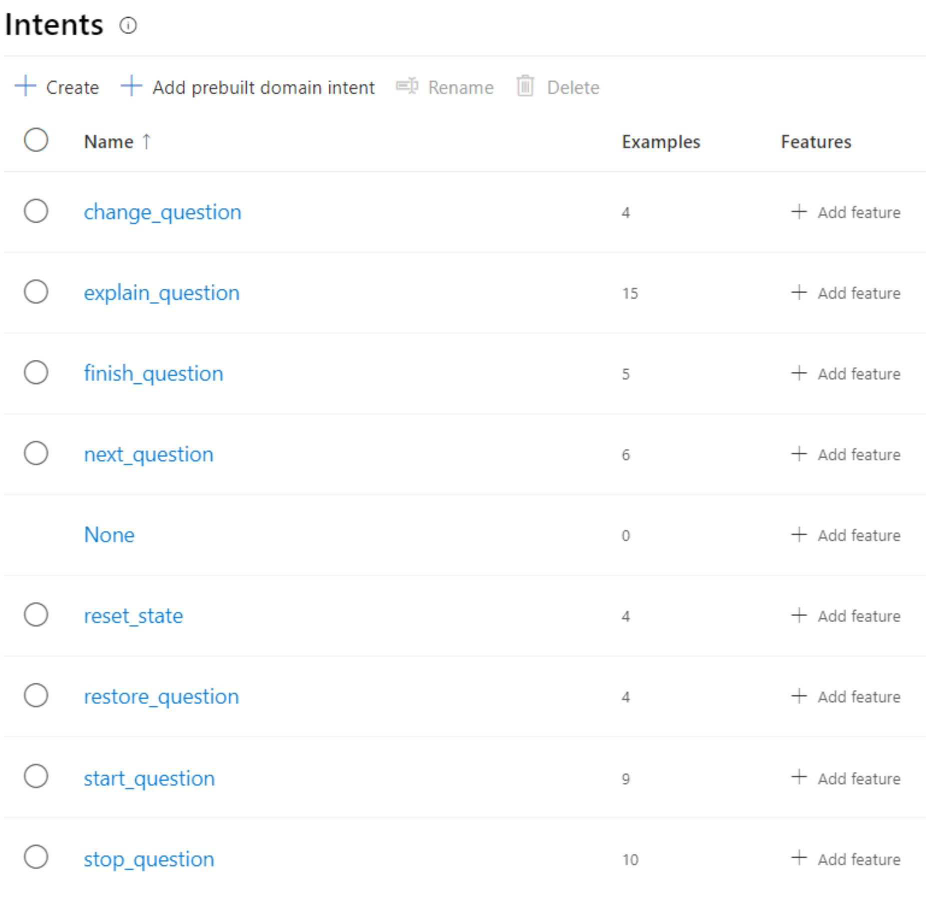


Fig. 4.5 Intent Classification on Microsoft LUIS Service

The design of the named entities of the educational robot tutoring homework system focuses on the names of the groups of questions. The groups of questions are used for students to select the corresponding questions. The name of the group can be the type of question or the chapter of the course. As long as the natural language understanding can recognize the group name, it can read the questions in the corresponding questions group of the database. The robotics tutoring math assignment configurator can configure the question group table by chapter assignments or by turning to a link. For example, as shown in Fig. 4.6

Fig. 4.6 Start_question Intent Template in Microsoft LUIS Service

in the intent recognition template, the named entity for the questions group name could be Chapter 3 Homework, Algebra.

4.3.3 Dialogue State Control Strategy

The conversation state control strategy of the conversation manager will be accomplished using a combination of finite state machines and the slot-filling method. Since the state of the educational robot tutoring math homework is relatively limited, it can be implemented using a finite state machine. Since the system needs to be obtained from user input. For the data-driven and neural network-based approaches, it is difficult to find the dataset for the application. Also, the project is inappropriate in application to children who are still mentally immature due to its inability to meet the output of safety. Since the system needs to select the set of question sets for the corresponding exercises according to the students' wishes, the system needs to collect slots to satisfy the switching conditions of the finite state machine, so the slot-filling method is applied here.

The first layer of the hierarchical finite state machine will map the state transition model of the educational robot tutoring math homework and implement the five states of the state transition mode—Question, Answer, Explain, Check, and Wait. The second layer of the finite state machine will effectively extend the sub-states implemented in the first layer of the finite state machine according to the system specific impleinment and the students' states.

The first layer of the finite state machine is shown in Fig. 4.7. the Question–Answer–Explain–Check state will complete the tutoring state cycle for a round of questions. the Wait state is used to handle situations where students are taking a break or have emergencies to attend to waiting for the robot tutoring system. In the Question, Answer, and Explain states, the robot provides important tutoring functions derived from the problem-solving model analysis. When a student does not hear clearly, the student can ask the robot to repeat the content. Since the question may appear to be beyond the student's ability, the student can ask the robot system to change the question or skip it. This reflects the limits of the freedom of ethical state transitions in the model implementation.

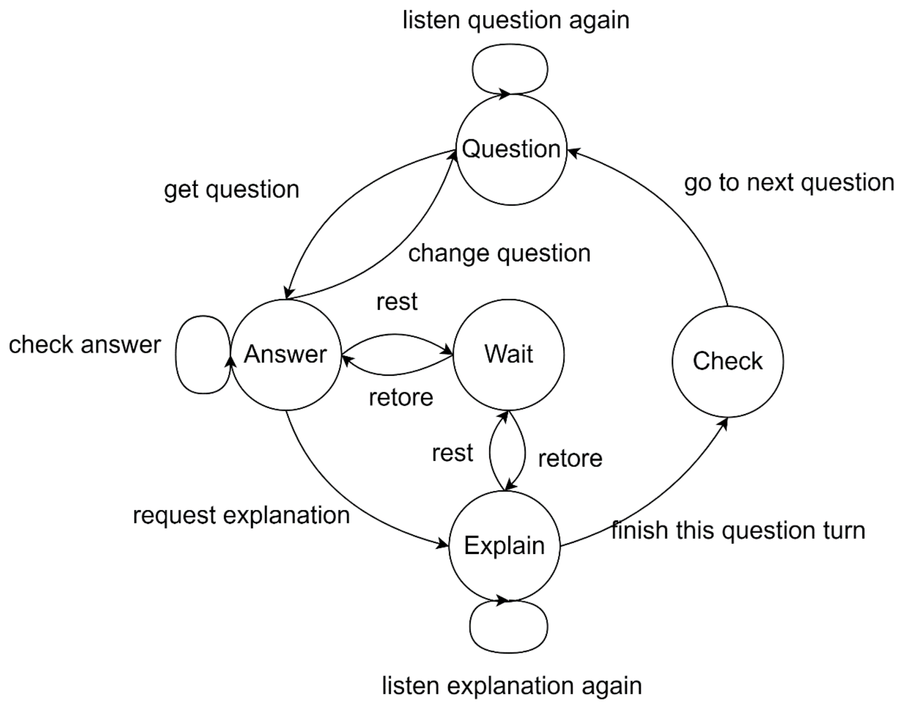


Fig. 4.7 First Layer Finite State Machine

The second layer of finite state machine is shown in Fig. 4.8. Some model details are omitted due to the cost of system implementation and the fact that the system is primarily a test of whether the basic state transition model can effectively perform the task of tutoring math homework. In the Question state, including Start, Entity, Recommend, and Change sub-states, the Start state is used to start the whole finite state machine for solving the question, and to get the intention classification and named entity of the user's input voice text

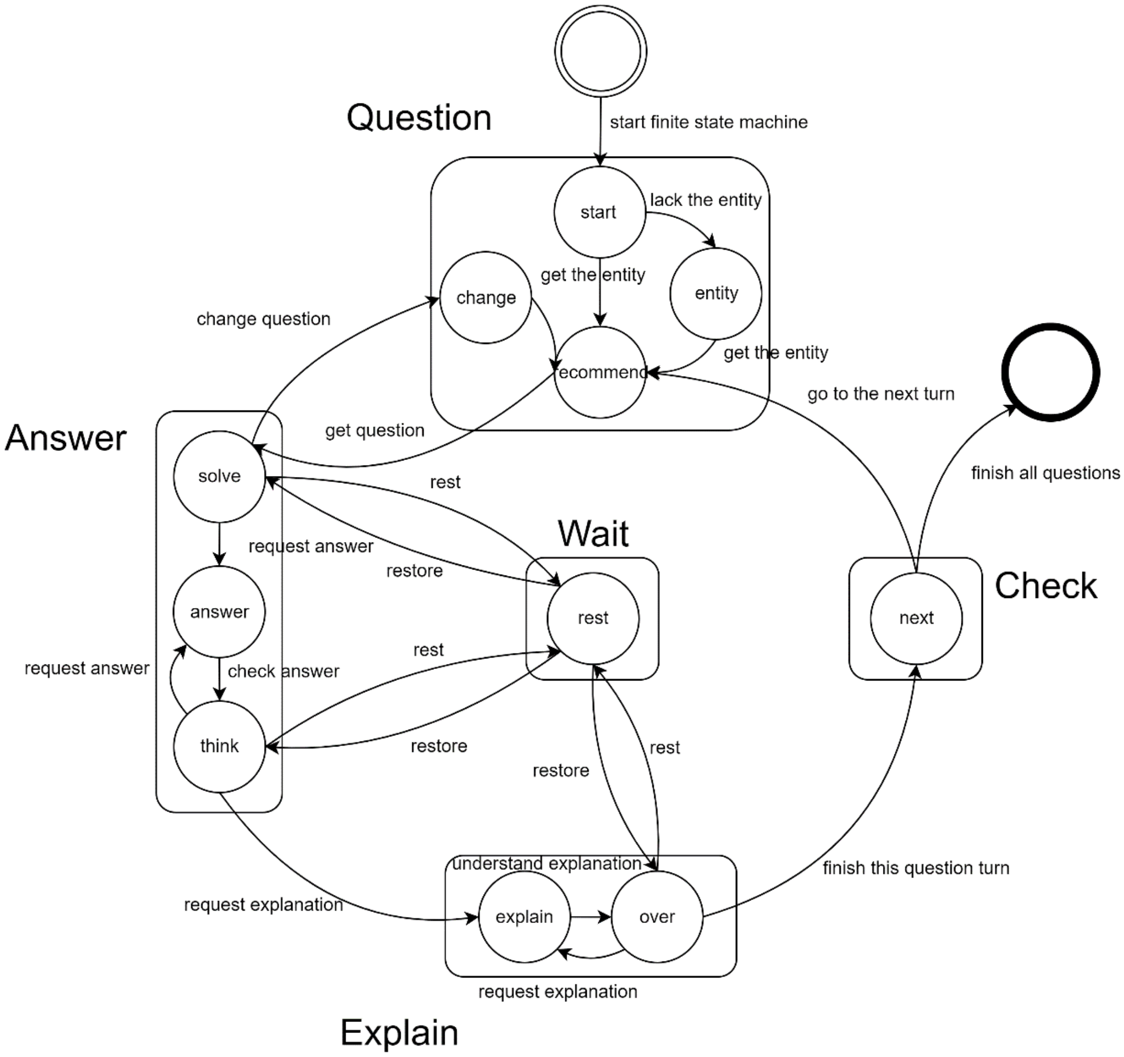


Fig. 4.8 Second Layer Finite State Machine

information. If the named entity is not obtained, the system switches to the Entity state and asks the user for the named entity about the entity. In the Recommend state, the program will get the questions with the corresponding group names from the question database to form the question list for this tutorial, and the robot system will tell students the question information from the first question The robot system will tell students the question information from the first question. The Change state accepts requests from the Solve state and will either regenerate the question in the Recommend state or use the current question list to skip the question. The Solve state is where the student is solving the question, the Answer state is where the robot helps the student check the answer, and the Think state is where the student thinks about the answer and his or her own solution and tries to solve the question again if the answer does not match. The Explain state is when the robot system explains the question to the student, and the Over state is when the student finishes solving the question and thinks about and understands the question. If it is completed, the finite state machine will end. If not it will go to the Next state transition round for tutorial math homework.

There are two ways to switch in the finite state machine, one way is to switch automatically when the system finishes execution. Another way is to switch triggered by the user's input of an audible message. The state in the former is the automatic state and the state of the latter is the continuous state. The automatic state is the locked state. It cannot be interrupted until the execution of the program is finished. The continuous state is not a locked state and can be interrupted. That is, only in the continuous state can the user input sound messages affect the operation of the finite state machine. Also, only during the persistent state does the machine process the user's input. The continuous state is actually the continuous problem-solving behaviour of the user student in completing several stages of the problem-solving model, such as solving and thinking.

In the second layer of the finite state machine, Entity, Recommend, Change, Answer, Explain, and Next states are automatic states. They can automatically switch to the later states after the robot tutoring system has finished the program in the state. In the automatic state, the finite state machine completes the tutoring task under each state in the state transition model as well as the cyclic implementation function of the system. As shown in Table. 4.4, the automatic states function table shows the relationship between the automatic state and the tutoring task of the subject and the system implementation.

The Start, Entity, Solve, Think, and Over states in the second level of a finite state machine are continuous states. Their switch will be influenced by the voice information input by the user. When the natural language understanding module gets the intention classification and entity information above the confidence, the finite state machine will switch to the corresponding state according to the user's intention. As shown in Table. 4.5, the table

Table 4.4 Automatic States Function Table

Parent State	Automatic state	Function
Question	Recommend, Change	Complete the generation of problem sets, as well as telling students about the question information.
Answer	Answer	Help students complete answer checking.
Explain	Explain	Explain the question to the student.
Check	Next	Reaching the next question tutoring status change to determine if all questions have been completed.
Wait	-	-

demonstrates intents and the function of intents and the relationship between the robot state transitions.

In the switching of Start and Entity states, it is necessary to obtain a slot to be able to switch. After the robot starts, the robot system enters the Start state and waits for user input. If the voice command entered by the user contains the slot content, i.e., the title group name. then the dialog system will directly switch to the Recommend state and execute the relevant program. If the voice command entered by the user lacks slot content. Then the dialog system will switch to the Entity state and ask for the relevant slot through the bot. For example, "Please tell me what type of question you want to do?" This design makes the interaction more natural.

4.3.4 Dialogue Output Control Strategy

The output control strategy of the Dialogue Manager is responsible for controlling the output of the Dialogue Manager. After the user state manager verifies the user's identity, the dialogue manager performs three layers of processing logic for voice text messages that match the user's identity using a chain-of-responsibility design pattern. The first layer is the detection of the confidence level for the intention classification in the natural language understanding model. Intentions detected above the confidence value are processed in the next layer. Intentions detected below the confidence level are responded to and the student is asked to re-enter the speech command. The second layer is the detection of the reachability of the finite state machine. The next level of processing is performed for reachable states. For unreachable states, a reply is generated and the student is asked to re-input the voice command. The third layer is for the execution of the execution procedure in the current state. The state switches in each intent are handled, as well as the reply templates for each intent using reflection. Reflection uses post-binding methods to detect the function methods

Table 4.5 Dialogue Manager Intent and State Transition Design

Intents	Intention Function.	Robot State Transition
start_question	Request to start question tutoring.	The robot enters the Question state, and when the robot completes the question state, it will enter the Answer state
change_question	Request the robot to change another math question.	The robot jumps from the Answer state to the Question state, reselects a question, and jumps back to the Answer state.
finish_question	Request the robot to perform an answer checking function. ◦	The robot state will be jumped inside the Answer state.
stop_question	Request the robot to enter the wait state.	The robot goes from the Answer state and the Explain state to the Wait state. and stores the current state.
restore_question	Request the robot to resume the current service.	The robot reads the previous state from the stored state. The robot reverts from the waiting state to the previous state.
explain_question	Ask the robot to perform a question explanation.	The robot jumps from the answer state to the Explain state.
next_question	Ends the tutoring cycle for the current turn.	The robot will end the tutoring for this turn. Check if the tutoring task is completed for all questions. If the tutoring is completed, the student will be praised and the tutoring session will end. If the tutoring is not completed, it will move to the next question in the tutoring status transition cycle.
reset_question	When an unknown error occurs, restart the question tutoring.	The robot restarts the question tutoring.

contained in the program set and call them at runtime. It can reduce the coupling of the program and optimize the implementation of the design pattern. A multi-layer processing and response system design based on the chain-of-responsibility design pattern [25] and reflection is shown in Fig. 4.9.

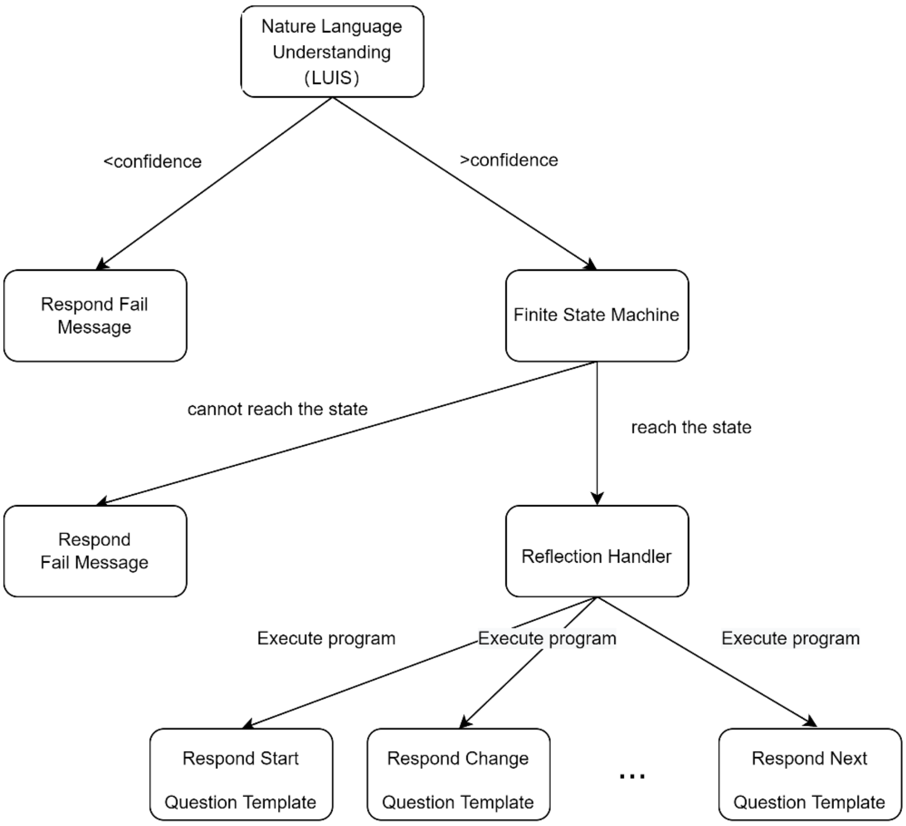


Fig. 4.9 Multi-Layer Processing and Response System Design Based on Chain-Of-Responsibility Design Pattern and Reflection

4.4 Multi-Channel Collaborative Control System

Unlike other dialogue managers that only output the response content, the dialogue manager in the educational robot tutoring math homework system outputs the robot not only the response content that requires speech synthesis, but also the control commands needed about the robot action system and the LED system. Through combination of the action, LED, the robot can express emotions during voice interaction[28]. This requires a cooperative control system for the multi-channel output of the robot. Two types of multi-channel cooperative control systems are summarized here. And one is selected to give the implementation.

4.4.1 Collaborative Control System Based on Intermediate Language Mapping

The intermediate language based cooperative control system uses XML markup text as an intermediate language to record the action sequences of the action modules of the robot cooperative control and the status commands of the LED system. And the XML intermediate markup language is used to map the hardware sensors so that the XML parser can execute the hardware actions represented by the intermediate language at the robot side based on the records of the intermediate language.

After the educational robot receives the JSON string output from the dialogue manager, it parses the part of the JSON string about the speech function by parsing the JSON string for speech synthesis; and passes the XML intermediate language tagging description of the cooperative control to the cooperative action parser. The collaborative action parser will parse the XML description language to complete the coordinated control of the robot's actions during the interaction. A simplified version of the robot cooperative control markup language is as follows.

```xml
<?xml version="1.0"?>
<xml>
  <sequence>
    <pose>
      <led1>1</led1>
      <led2>1</led2>
      <led3>1</led3>
      <LeftArmServo1>50</LeftArmServo1>
      <LeftArmServo2>20</LeftArmServo2>
      <LeftArmServo3>70</LeftArmServo3>
    </pose>
    <time>2000</time>
    <voice>Hello, Tom</voice>
  </sequence>
  <sequeue>
    <pose>
      <led1>0</led1>
      <led2>0</led2>
      <led3>0</led3>
      <LeftArmServo1>0</LeftArmServo1>
      <LeftArmServo2>0</LeftArmServo2>
      <LeftArmServo3>0</LeftArmServo3>
    </pose>
    <time>1000</time>
  </sequeue>
</xml>
```

Fig. 4.10 Robot Cooperative Control Markup Language Example

The XML token script records the posture information of each timepiece of the robot's collaborative control system, as well as the time held during that time period. As shown in Fig. 4.10, <sequence> represents the information of a timepiece of the robot, and the <pose> label stores the position information of the robot in each timepiece, including motion information and LED information. The current posture maintenance time is stored in the label <time>. The information that the current timepiece will pass to the speech synthesis module is recorded in the label <voice>. When the time is over, the robot will adjust the posture to the next position. In Fig. 4.10, the transformation between the two postures is recorded. Three sets of LED lights will be turned on. The Three servos of the left arm will rotate at an angle. This posture is maintained for 2000ms. The context in the label <voice> is passed to the voice module for speech synthesis operations. When the time is over, the robot will move into the posture in the next timepiece. The Three groups of LED lights will be turned off. The Three servos on the left arm will be restored to their original state. This posture will last for 1000ms. Since no voice information is required for speech synthesis, this timepiece does not need to pass a message to the voice module.

However, this implementation is extremely complex and requires a lot of time and cost to label each timepiece. For junior programmers and non-professionals such as teachers, it is difficult to complete the design of movements. In addition, because the edited action posture is designed to the bottom of the hardware, it is likely that the edited action will cause the robot to fall or the robot to damage each hardware module. Two papers have been given to this realization[29], [30]. This article will take the second implementation method to give the implementation.

4.4.2 Collaborative Control System Based on Predefined Actions

Because the HAL layer of the robot operating system abstracts the hardware layer of the robot, it provides the most basic class libraries of the robot. Even some robot models provide visual editing tools. So with the HAL layer, basic action groups can be designed and stored in the robot operating system. So this provides an easy multi-channel motion co-control system for robots. By pre-defining the basic actions in the robot operating system, the complex actions are combined and stored in the robot operating system and waiting for the program to call. As shown in Fig. 4.11, simple actions and LED states are combined into a continuous complex state in the NAO robot visual editor software. In the Edu_action project as well as to complete the editing of 4 groups of actions.

After receiving the return JSON message from the dialogue manager, the response manager first processes the return message and extracts the fields needed for speech synthesis to send to the speech synthesis module. The speech module will call the robot's speech

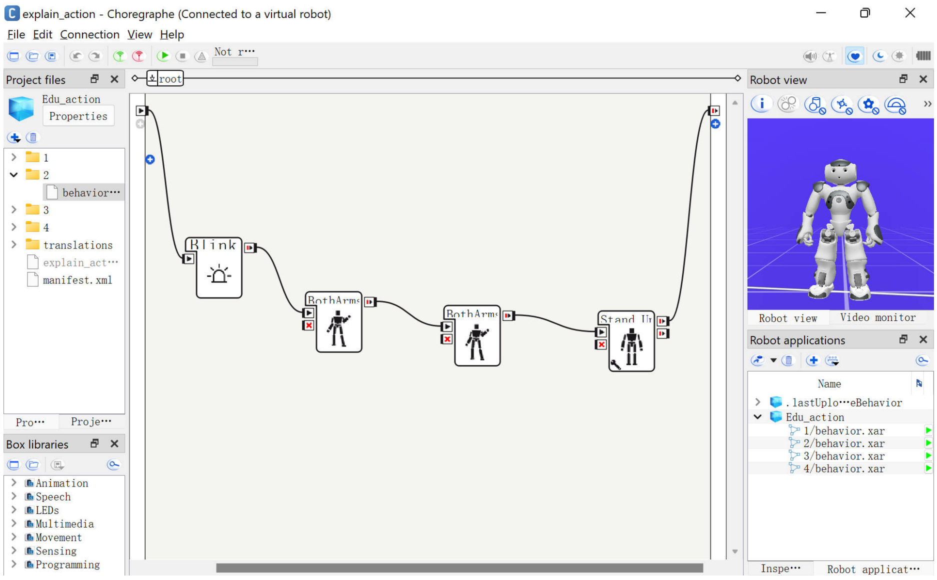


Fig. 4.11 Choregraphe Software Action Design Interface

synthesis interface for speech output. Then, the respond manager sends the action group tags to the action tag manager. The action tag manager will invoke the predefined actions stored in the robot operating system based on the action group tags. Fig. 4.12 shows the processing flow of the robot multi-channel cooperative control system based on predefined actions.

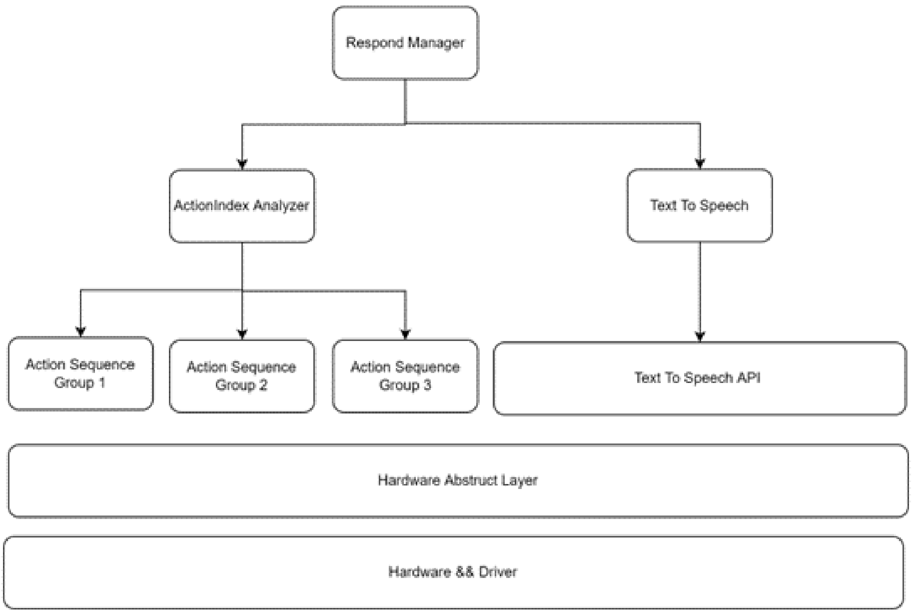


Fig. 4.12 Co-control System Processing Flow Based on Predefined Actions

This multi-channel cooperative control system leverages the hardware abstraction layer provided by the robot system to implement complex movements by combining simple actions. This approach makes it easy for non-specialists to design the robot's movements and LED states during the interaction. Meanwhile, there is no need to worry about the robot falling down due to unreasonable motion design or hardware damage due to unreasonable programming.

4.5 Summary

The main content of this chapter is the system implementation of the state transition model. This chapter introduces the system framework of the educational robot tutoring homework system, the cloud service support system, the dialogue manager, and the robot multi-channel collaborative control system. The cloud service support system includes question database, log database, and teacher management tool. The dialogue manager includes dialogue dominant strategy, dialogue input strategy, dialogue state control strategy, and dialogue output control strategy. The multi-team collaborative control system includes two types of collaborative control methods. Collaborative control system based on intermediate language mapping and collaborative control system based on predefined actions.

Chapter 5

Experiment

5.1 Experiment Environment

The thesis uses the NAO robot as the experimental platform. NAO robot is an automated humanoid robot developed by the French robotics company Aldebaran Robotics. The Nao robot is controlled by a linuxbased operating system called naoqi, which was acquired by Softbank and renamed Softbank Robotics. The operating system has a built-in interface for controlling the robot's behaviour, voice, LED and other functions. It includes four microphones, two speakers, two HD cameras, and two sets of sonar sensors. Since the recognition rate of NAO robot's own speech recognition system is relatively poor, this thesis uses Microsoft speech recognition service as the speech recognition part of the robot. In addition, since voice recognition is difficult to implement in the NAO robot, the experiments related to voice recognition in this thsis assume that this part has been completed and uploaded to the identification of the identified person in the multi-round dialogue system. As shown in Fig. 5.1 , a real-time printout of the NAO robot program on the computer is being tested in the robot's silent state for real-time recording of the robot's interactions in tutoring students with their math homework. As shown in Fig. 5.2, the NAO robot is being used to tutor students on their math homework. And its interaction process will be printed in real time on another computer for evaluating its interaction process.

The multi-round dialogue system will have a C# program implementation, deployed to the Web App Service application service in Azure Server using the WEB API5 framework in ASP.NET to provide REST API services. The education bot will send HTTP requests to the WEB API server via python to receive the returned answers. The HTTP requests from the education bot will trigger changes in the interaction system so that intent and entities are identified to the LUIS Language Understanding Service based on the uploaded text

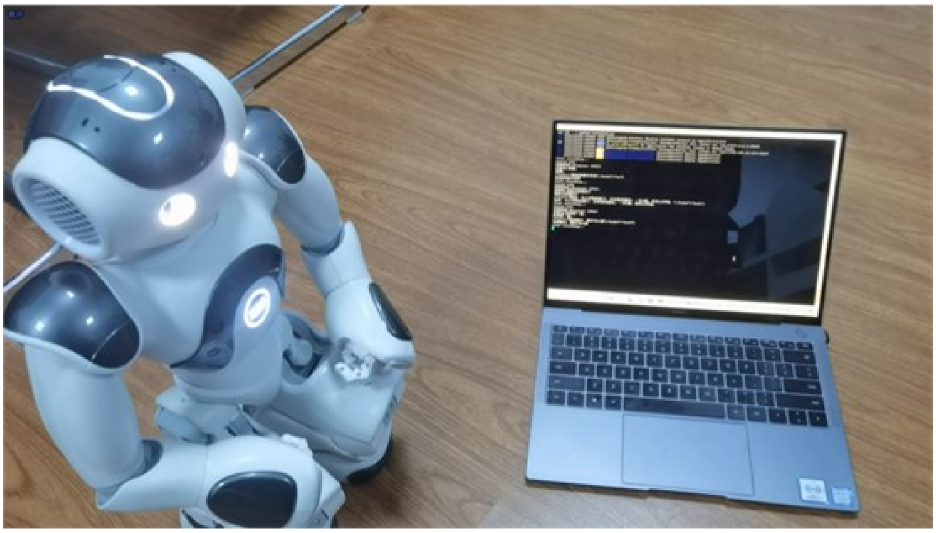


Fig. 5.1 Real-time Printout of NAO Robot

Fig. 5.2 Nao Robot Tutoring Homework

strings, which drive state transition in the finite state machine. The user state manager plays a supporting role in this process.

The cloud service support system contains a problem database, log database, and instructor tool pages to be deployed on Azure cloud servers. The question database and log database will be served using relational services in the Azure server. The teacher tool related pages will be deployed in the same Web App Service application service along with the WEB API5 framework.

5.2 Robot State Transition Experiment

The Table. 5.1 is a real-time printout from the robot console while the educational robot tutoring math homework system is running. The printout is processed to select the round of interactions in the interaction and to remove the failed interaction content due to failed speech recognition content. The user requests the robot to start tutoring math homework. The robot finishes executing the automatic state in the first Question state and stays in the sub-state Answer in the Question state Answer state, which is the Work state in the print. Then the user asks for a break and the robot system jumps to the Rest state, a sub-state of the Wait state. After finishing the break, the user asks to resume the tutoring activity and the robot system resumes from the sub-state Rest state of the Wait state to the sub-state Answer state. After the user finishes solving the problem, the robot is asked to check the answer. After the robot completes the question checking automatic state, it jumps within the Answer state and reaches the Think state. After thinking about the question again, the user asks the robot to explain it, and the robot enters the Explain state of the Explain question automatic state and reaches the Over state inside the Explain state after completing it. In Over the user will digest the knowledge behind the question. When the user wants to start the next question, the robot system will enter the Check state and execute the automatic state within the Check state, thus completing this round of tutoring and starting the next round of Question state, which tells the user about the next question.

The actual situation of the educational robot tutoring math homework above shows that the robot can complete the task of tutoring math homework after five state transitions. In the five state transitions, the tutoring task in the problem-solving model is completed. Thus, it confirms that the state transitions of the state transition model can complete the tutoring task of math homework.

Table 5.1 Robotic Tutoring Homework System Console Prints Records

Turn	Print
1	start recording...start recording...record over 识别请求:<Response [200]> 识别为:想要做几何题目? {"info":"你好，马上开始做题了。正在推荐题目。一共4题，请从12开始。","actionIndex":"2","model":"answer"}
2	start recording... record over 识别请求:<Response [200]> 识别为:想要休息一下? {"info":"你可以休息了，记得回来啊","actionIndex":"0","model":"rest"}
3	start recording... record over 识别请求:<Response [200]> 识别为:回来了。 {"info":"欢迎回来哦","actionIndex":"0","model":"answer"}
4	start recording... record over 识别请求:<Response [200]> 识别为:做完了? {"info":"第12题答案是周长为5","actionIndex":"0","model":"think"}
5	start recording... record over 识别请求:<Response [200]> 识别为:给我讲解一下。 {"info":"三角形的周长等于三边相加","actionIndex":"1","model":"over"}
6	start recording... record over 识别请求:<Response [200]> 识别为:等下一批。 {"info":"请重复一下命令","actionIndex":"0","model":"over"}
7	start recording... record over 识别请求:<Response [200]> 识别为:议题。 {"info":"请重复一下命令","actionIndex":"0","model":"over"}
8	start recording... record over 识别请求:<Response [200]> 识别为:开始下一题。 {"info":"下一道题目为第13题","actionIndex":"3","model":"answer"}

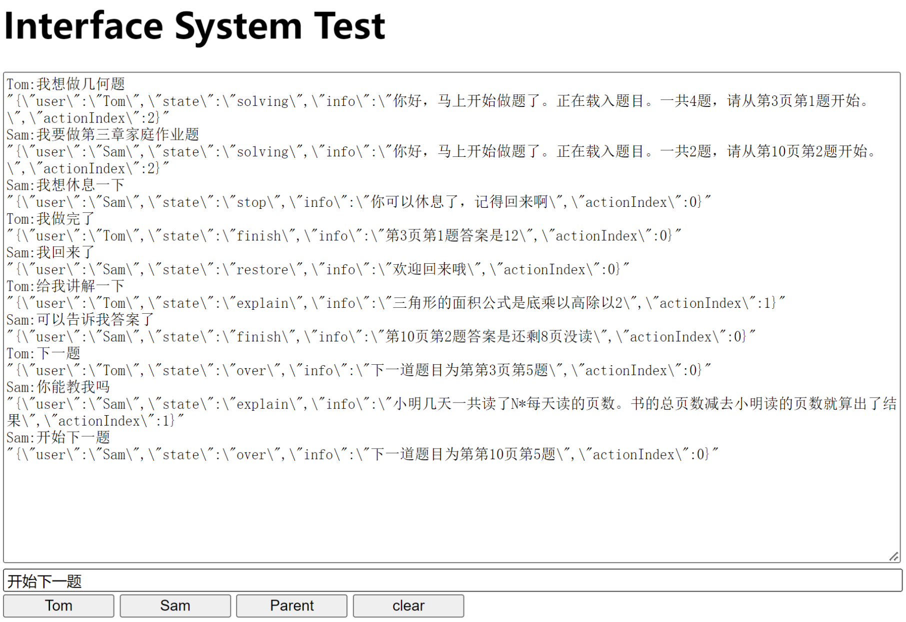


Fig. 5.3 Robot Tutoring Multiple Students in Experiments

5.3 Robot Tutoring Multiple Students Experiment

As described in the experimental setting, since voice recognition is difficult to implement on the NAO robot, it is assumed that voice recognition has been completed by passing a request for a multi-round conversation manager. As shown in Fig. 5.3, students Tom and Sam each interacted with the interaction system using their respective identifiers and states, sharing a finite state machine structure implemented by the state transition model, to complete their respective math homework tutoring activities. They each completed state transitions about the robotic system tutoring math homework. It is demonstrated that the robot can tutor multiple students at the same time. It is also verified that the state transition model can be adapted to the case where the robot tutors multiple students at the same time.

5.4 Robot Tutoring Parent-Accompanied Students Experiment

As shown in the Fig. 5.4, two students and two parents are engaged in educational robot home tutoring together. The students do not understand the explanation after listening to the robot, and the parents help the students to understand the explanation of the question after understanding the explanation. In the first processing method, the parents' voice input is treated as noise, and all the students drive the state transition of the robot tutoring homework

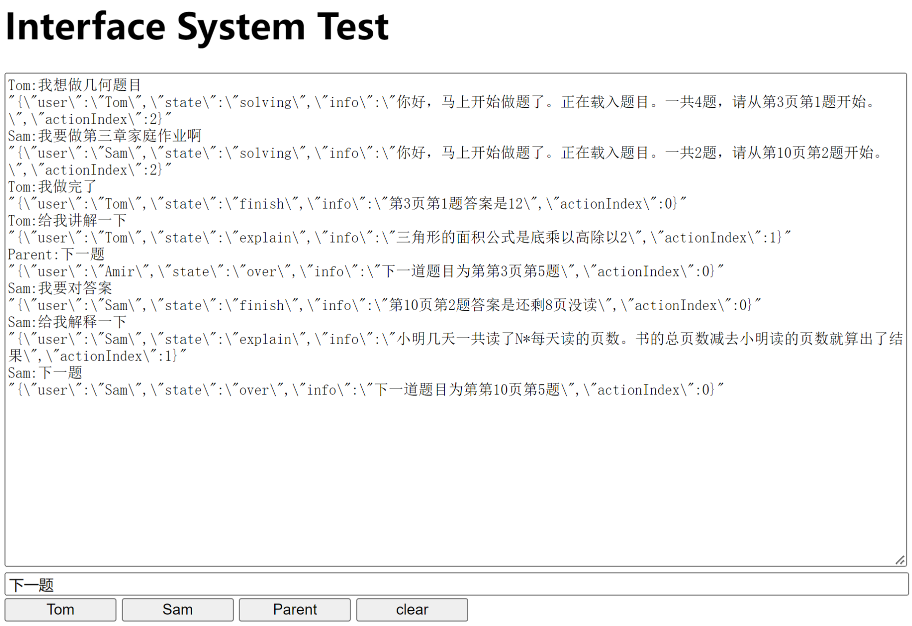


Fig. 5.4 Robot Tutoring Parent-Accompanied Students in Experiments

system. In the second processing method, the parent can drive the robot to the next question after explaining the question to the student again, which is a more natural way of interaction. From the above experiments, it can be verified that the state transition model of educational robots tutoring homework can be adapted to the case of robots tutoring students accompanied by parents.

5.5 Teacher Guiding Robot Experiment

Teachers can configure the question database through the Teacher Tools page. As shown in Fig. 5.5, teachers add new questions and question groups. The image after adding is shown in Fig. 5.6. The robot system runs in the Question, Answer, and Explain states, extracts the question tutoring task-related states from the database attributes Question, Answer, and Explanation, respectively, and performs the tutoring task in each state. The teacher directly influences the operation of the robot Question, Answer, Explain state by configuring the question database. In addition, the number of questions under the current question group affects the list of questions in the robot system when it is running, and indirectly affects the Check state, which is responsible for the transition of question rounds. The Check state records the transition of the robot system state, and in turn provides the teacher with a means to observe student learning. The log database, shown in Fig. 5.7, illustrates the robot tutoring

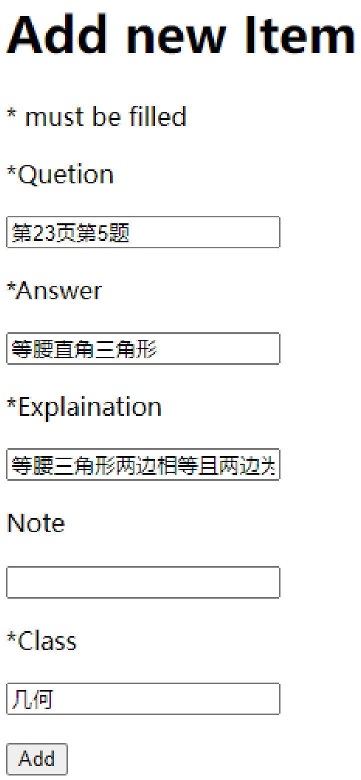
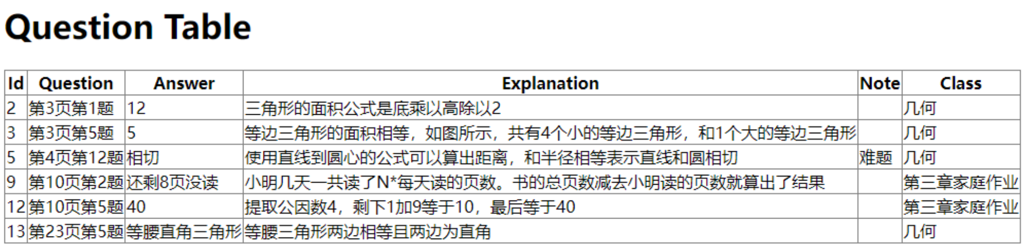


Fig. 5.5 Teacher Adding Questions Page


Question Table

Id	Question	Answer	Explanation	Note	Class
2	第3页第1题	12	三角形的面积公式是底乘以高除以2		几何
3	第3页第5题	5	等边三角形的面积相等，如图所示，共有4个小的等边三角形，和1个大的等边三角形		几何
5	第4页第12题	相切	使用直线到圆心的公式可以算出距离，和半径相等表示直线和圆相切	难题	几何
9	第10页第2题	还剩8页没读	小明几天一共读了N*每天读的页数。书的总页数减去小明读的页数就算出了结果		第三章家庭作业
12	第10页第5题	40	提取公因数4，剩下1加9等于10，最后等于40		第三章家庭作业
13	第23页第5题	等腰直角三角形	等腰三角形两边相等且两边为直角		几何

Fig. 5.6 Teacher Questions View Page

students through two math assignments. The Finish state here refers to the sub-state Answer under the Answer state.

Log Table

Id	User	State	Time
1	Tom	start	6/03/2022 8:42:26 PM
2	Tom	recommend	6/03/2022 8:42:27 PM
3	Tom	solving	6/03/2022 8:42:28 PM
4	Tom	finish	6/03/2022 8:42:51 PM
5	Tom	think	6/03/2022 8:42:51 PM
6	Tom	explain	6/03/2022 8:43:00 PM
7	Tom	over	6/03/2022 8:43:00 PM
8	Sam	start	6/03/2022 8:43:59 PM
9	Sam	recommend	6/03/2022 8:43:59 PM
10	Sam	solving	6/03/2022 8:44:00 PM
11	Sam	finish	6/03/2022 8:44:18 PM
12	Sam	think	6/03/2022 8:44:19 PM
13	Sam	explain	6/03/2022 8:44:34 PM
14	Sam	over	6/03/2022 8:44:34 PM

Fig. 5.7 Log Database View Page

The teacher-guided robot experiment demonstrates how the teacher is involved in the robot's activity of tutoring homework. Teacher involvement provides the three task states in the state transition model that arise from the problem-solving model. The experiments validate that the robot tutoring homework model transition design can be adapted for teacher involvement in robot tutoring homework.

5.6 Summary

The main content of this chapter is to verify the validity of the model through practical system testing. This chapter states the experimental environment and the results of four experiments. This chapter first tests the educational robot tutoring a single student's math homework activity in a real NAO robotics environment to verify that the state transition model can complete the student's math homework tutoring by jumping through states. Secondly, because of the difficulty of the NAO platform vocal implementation, it is verified that the model can be adapted to both scenarios through state management by printing interactive statements in the scenarios of parental involvement in tutoring activities and multi-student involvement in tutoring activities. Finally, it can be demonstrated that teachers can be indirectly involved in the transformation of the state transition model by showing the configuration of the problems in the teacher participation state transition model.

Chapter 6

Conclusion and Future Work

In this thesis, we design a state transition model for educational robots tutoring math homework. The model fully considers the problem-solving model, the system implementation, and the individual state requirements of students, using five states to outline the states needed for multiple rounds of interaction for educational robots tutoring homework. It also discusses the multiple roles involved in robot tutoring math homework, and the ethical constraints to be aware of when implementing the system. The paper implements an educational robot tutoring homework system based on the model. The system consists of a multi-round dialogue manager based on hierarchical state machines and natural language understanding and a multi-channel collaborative control system for robots, and experiments are conducted on an NAO robot. It is verified that the state transition model of educational robot tutoring homework can generalize the situation of robot tutoring math homework, while the model can adapt to the situation of multiple roles involved in robot tutoring homework activities.

This thesis solves the multi-turn interaction problem and the multi-role participation problem in educational robot tutoring homework through the state transition model, and verifies the effectiveness of the model through experiments. The next step is to actually test the robot tutoring system in a real-world environment. The states in the state transition model and the implementation of the states can be further optimized and improved in actual use. On the other hand, mathematics is the foundation subject of science subjects, and the problems of science subjects and mathematics problems have commonality. Therefore, it is worthwhile to try whether the state transition model based on the math problem-solving model can be modified appropriately and applied to other science subjects.

References

[1] J. Shen, Y. Yin, L. Li, L. Shang, X. Jiang, M. Zhang, and Q. Liu, “Generate & Rank: A Multi-task Framework for Math Word Problems,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2269–2279.

[2] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, “Training Verifiers to Solve Math Word Problems,” *arXiv:2110.14168 [cs]*, Nov. 2021.

[3] C. C. Federation, “Primary School Math Application Problems Automatic Problem Solving Competitions(in chinese),” https://www.datafountain.cn/competitions/467/ranking?isRedance=0&sch=1709, Mar. 2022.

[4] Ministry of Education of the People’s Republic of China, “Strengthen the management of mobile Internet applications in education Promote convergence with the "double reduction" policy(in chinese),” http://www.moe.gov.cn/jyb_xwfb/gzdt_gzdt/s5987/202112/t20211213_586971.html, Feb. 2022.

[5] ——, “The General Office of the CPC Central Committee and the General Office of the State Council issued the Opinions on Further Reducing the Burden of Homework and Off-campus Training for Students in the Compulsory Education Stage(in chinese),” http://www.moe.gov.cn/jyb_xxgk/moe_1777/moe_1778/202107/t20210724_546576.html, Mar. 2022.

[6] Taoyun Technology, “Alpha Egg Robot Product(in chinese),” http://www.toycloud.com/channels/198.html, Sep. 2021.

[7] Huawei, “Video of the Whole House Intelligence and Intelligent Screen Flagship New Product Launch(in chinese),” https://consumer.huawei.com/cn/press/video/2021/v-2021/.

[8] X. Yan and s. Peng, “Algorithm Design and Implementation of Multi-turn Chatbot in Education,” *Software Engineering(in chinese)*, vol. 24, no. 02, pp. 30–33, 2021.

[9] P. Vogt, M. de Haas, C. de Jong, P. Baxter, and E. Krahmer, “Child-Robot Interactions for Second Language Tutoring to Preschool Children,” *Frontiers in Human Neuroscience*, vol. 11, p. 73, Mar. 2017.

[10] J. Leoste and M. Heidmets, “Bringing an Educational Robot into a Basic Education Math Lesson,” in *Robotics in Education*, M. Merdan, W. Lepuschitz, G. Koppensteiner, R. Balogh, and D. Obdržálek, Eds. Cham: Springer International Publishing, 2020, pp. 237–247.

[11] J. Shim, D. Kwon, and W. Lee, “The Effects of a Robot Game Environment on Computer Programming Education for Elementary School Students,” *IEEE Transactions on Education*, vol. 60, no. 2, pp. 164–172, May 2017.

[12] L. Brown and A. M. Howard, “Engaging children in math education using a socially interactive humanoid robot,” in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. Atlanta, GA: IEEE, Oct. 2013, pp. 183–188.

[13] R. B. Rosenberg-Kima, Y. Koren, and G. Gordon, “Robot-Supported Collaborative Learning (RSCL): Social Robots as Teaching Assistants for Higher Education Small Group Facilitation,” *Frontiers in Robotics and AI*, vol. 6, p. 148, 2020.

[14] H. Chen, H. W. Park, and C. Breazeal, “Teaching and learning with children: Impact of reciprocal peer learning with a social robot on children’s learning and emotive engagement,” *Computers & Education*, vol. 150, p. 103836, 2020.

[15] S. Matsuura and R. Ishimura, “Chatbot and Dialogue Demonstration with a Humanoid Robot in the Lecture Class,” in *Universal Access in Human–Computer Interaction. Human and Technological Environments*, ser. Lecture Notes in Computer Science, M. Antona and C. Stephanidis, Eds. Cham: Springer International Publishing, 2017, pp. 233–246.

[16] M. Satar, “Speaking with Machines: Interacting with Bots for Language Teaching and Learning.” *Research-publishing. net*, 2021.

[17] T. T. Nghi, T. H. Phuc, and N. T. Thang, “Applying ai chatbot for teaching a foreign language: An empirical research,” *Int. J. Sci. Res*, vol. 8, 2019.

[18] J.-X. Huang, K.-S. Lee, O.-W. Kwon, and Y.-K. Kim, “A chatbot for a dialogue-based second language learning system,” in *CALL in a Climate of Change: Adapting to Turbulent Global Conditions – Short Papers from EUROCALL 2017*, K. Borthwick, L. Bradley, and S. Thouësny, Eds. Research-publishing.net, Dec. 2017, pp. 151–156.

[19] S. Ruan, A. Willis, Q. Xu, G. M. Davis, L. Jiang, E. Brunskill, and J. A. Landay, “BookBuddy: Turning Digital Materials Into Interactive Foreign Language Lessons Through a Voice Chatbot,” in *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*. Chicago IL USA: ACM, Jun. 2019, pp. 1–4.

[20] G. Polya, *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press, 2004.

[21] A. Yimer and N. F. Ellerton, “A five-phase model for mathematical problem solving: Identifying synergies in pre-service-teachers’ metacognitive and cognitive actions,” p. 17, 2010.

[22] J. D. Williams and S. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.

[23] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, "The Design and Implementation of XiaoIce, an Empathetic Social Chatbot," *Computational Linguistics*, vol. 46, no. 1, pp. 53–93, Mar. 2020.

[24] E. Gamma, R. Helm, R. Johnson, R. E. Johnson, J. Vlissides *et al.*, *Design patterns: elements of reusable object-oriented software*. Pearson Deutschland GmbH, 1995.

[25] S. Vinoski, "Chain of responsibility," *IEEE Internet Computing*, vol. 6, no. 6, pp. 80–83, 2002.

[26] C. S.-S. Cheung and E. M. Pomerantz, "Parents' Involvement in Children's Learning in the United States and China: Implications for Children's Academic and Emotional Adjustment: American and Chinese Parents' Involvement," *Child Development*, vol. 82, no. 3, pp. 932–950, May 2011.

[27] E. M. Pomerantz, E. M. Kim, and C. S.-S. Cheung, "Parents' involvement in children's learning." 2012.

[28] R. Hu, X. Yu, B. He, and Z. Li, "Implementation of voice interaction based on questions and answers for guide robot in exhibition hall," in *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, Dec. 2017, pp. 105–109.

[29] H. Liu, "Task-oriented Control System and Programming Method for Educational Humanoid Robot(in chinese)," M.S. thesis, Shanghai Jiao Tong University, 2013.

[30] M. Yang, "Nature Multimodal Human-Computer-Interaction Dialog System," *Computer Science(in chinese)*, p. 8, 2014.

Chinese Abstract

最近OpenAI和北京大学数学系的研究人员使用新方法将自动解题的准确率提高，OpenAI使得GT3模型在解答小学数学题中的水平提高至55分。在2020年中国计算机学会举办的自动解题赛事中，这个解答的正确率还是40%左右。虽然自动解题自上世纪60年代就开始研究，但取得的总体准确率效果依然不高。最近几年的研究有显著进步，可解释化自动解题引擎已经可以解决一部分简单的数学题目。未来随着科技进步，可解释性的自动解答引擎将逐渐进化成为可靠和实用的人工智能功能。为了充分发挥自动解答数学题引擎的作用，我们需要将教育机器人与其相结合形成教育机器人辅导数学家庭作业的应用。基于此，教育机器人将可以承担起家庭教育场景中，儿童和教师沟通的桥梁，并代替人工给儿童进行讲解题目。

另一方面，由于互联网和大数据的发展，题拍拍等题目拍照搜题软件造成了大量的学校教育问题。儿童大量使用软件搜索题目答案来代替自我思考。这种情况使得教育部下发命令下架题拍拍模式的软件。配合中国教育部门之前开展的尝试减轻学生重复的课业负担的政策，降低作业时长和作业量，提升作业质量。在双减问题的大环境下，家庭中辅导作业的教育机器人可能是一个解决题拍拍等遗留的问题好途径，并保护学生避免受到人工智能的侵害，将人工智能在教育方面的应用放置于教师和专家的管控之下。教育机器人将有效避免提拍拍作业辅导软件所带来的负面影响，它能够智能的完成对儿童进行家庭作业辅导的任务，在一定程度上拯救了缺乏教学经验，而精疲力尽的家长。

市场上已经有很多家庭辅导机器人，但是它们更像是具有机器人外表的故事机器和视频播放器。这些机器人仅仅具备单轮交互的能力。然而在教育机器人数学家庭辅导的过程中，教育机器人需要辅助学生完成数学解题的过程。这是一个复杂的、具有多轮交互的过程。另一方面，随着语义理解技术的成熟，多轮对话系统开始逐渐应用在各种领域中。教育领域的研究人员也对多轮技术进行了相关的研究，他们将多轮对话系统应用于口语训练，课堂知识问答，辅助学习数字材料等方面等场景。但是仍然缺乏对于教育机器人辅导数学家庭作业多轮交互过程的研究。为了实现教育机器人辅导数学家庭作业的过程，机器人需要完成很多状态的转换。这些状态需要能够完成辅导数学家庭作业的过程中的辅导任务，同时满足系统的实现，以及用户的个人需求。同时，通过这些状态的变换，教育机器人能够完成辅导家庭

作业的多轮交互过程。另一方面，机器人辅导数学家庭作业可能面临着多种角色参与家庭辅导的情况。这需要教育机器人的状态转换模型的状态转换管理能够适应这些情况，使得教育机器人具有同时辅导多名学生和辅导家长陪伴下的学生的功能。另外，教师应当能够干预教育机器人的辅导过程，从而指导教育机器人的家庭辅导，以及间接和儿童建立关联，参与到机器人辅导家庭作业的过程中来。

论文假设的场景是教育机器人在家庭中承担辅导儿童数学家庭作业的任务。辅导对象为8–12岁，具有一定自主思考和学习能力儿童。他们不需要教师手把手或是富有情感教导，可以接受与机器人维持长时间的交互，不会对其心智发展造成不利影响。教育机器人由专家和教师共同管控，教师负责对机器人的辅导活动进行教学指导，专家负责教育机器人的研究和学生交互行为的分析。

论文以NAO机器人作为研究的平台。针对教育机器人家庭辅导中存在的问题，论文设计了一个教育机器人辅导数学家庭作业的状态转换模型，用于解决机器人辅导家庭作业中的多轮交互问题。状态转换模型决定着机器人服务的质量。所设计的模型还考虑了在多种实体参与家庭辅导活动的情况下的状态管理的合理性。经过对模型实现的教育机器人辅导数学家庭作业系统的测试，验证模型可以完成机器人辅导数学解题作业的多轮交互任务，同时模型在多角色参与机器人辅导家庭作业活动的情况下运作良好，即机器人同时辅导多名学生的情况，机器人辅导家长陪伴下的学生的情况，教师指导教育机器人的辅导活动的情况。论文的主要贡献如下。

1.论文设计了教育机器人辅导家庭作业的状态转换模型。

本文首先研究数学解题过程模型，确定学生在数学题目问题解答中的需求，得到了机器人辅导数学家庭作业的辅导任务状态。在充分考虑到系统实现需要的状态和学生个人需求的状态后，确定了教育机器人在家庭作业辅导中状态集合——问题，回答，讲解，检查，等待。论文设计了一个状态转换模型，涵盖状态集合中的所有状态。所设计的状态转换模型还考虑了在多种角色参与家庭辅导活动情况下的状态管理的合理性。状态转换模型可以适应教育机器人辅导多名学生的情况，教育机器人辅导家长陪伴的学生的情况，以及教师指导机器人辅导家庭作业活动的情况。并通过教育机器人辅导数学家庭作业系统验证了模型的有效性。

2.论文根据模型实现了一个面向家庭辅导活动的多轮对话管理器。

该对话管理器可以在辅导过程中实现模型中的状态变换，完成家庭辅导中的任务，并兼容辅导场景中的多种角色参与的情况。多轮对话管理器由用户和系统混合主导，学生的语音命令对辅导过程进行驱动，系统询问学生取得缺失的实体。多轮对话管理器使用分层有限状态机作为实现方法。模型的基本状态作为第一层有限状态机的状态，模型实现的细节以及学生解题思考过程，学生在系统中的模型转换过度状态作为第二层有限状态机。模型使用用户状态管理器存储每个不同的学生和家长的状态，每个学生建立单独的标志及其配置文件，从而实现教育机器人可以分辨多个学生的交互，从而实现模型对于教育机器人同时辅导多名学生情况的适应。对

话系统将家长视为一个特殊的交互对象，家长可以参与教育机器人辅导活动中的讲解，并用更适合学生的语言帮助学生理解机器人的，并控制教育机器人跳转到下一个状态。教师可以通过配置教育机器人生成问题的题库来进行间接参与教育机器人对于家庭辅导的过程。教师配置题目库是问题的集合，包含多道题目的目录、初步解析，以及题目的答案和题目的讲解。教师可以通过手动配置题目，也可以根据推荐系统进行题目选择以及未来的自动解题引擎。教师可以查看教育机器人记录的学生解题过程交互日志来观察学生实际的家庭作业完成情况，从而及时调整对于学生教学计划。

3.论文总结了两种教育机器人家庭辅导活动中多通道输出协同控制的设计方法。

在缺乏硬件抽象层的机器人环境，教育机器人实现可以通过XML中间语言，对多通道输出的动作序列、LED状态、语音回复内容进行标记。通过对XML记录的动作序列的解析，教育机器人实现在家庭辅导活动中的多通道协同输出。在拥有硬件抽象层的机器人环境，教育机器人可以预先设计动作序列组，动作序列组包括机器人动作序列以及机器人LED变化，教育机器人接受对话管理器传递的命令参数，并将其中回复内容进行语言合成输出，使用动作标签解析器对其中的动作序列进行解析，调用硬件抽象层中存储的动作序列组。论文选取第二种方法进行实现。这种实现可以通过可视化的动作编辑器以及现有的动作进行简单动作的组合输出。它可以使用硬件抽象层中已经拥有的动作接口，降低动作设计和解析成本。另一方面，这种方法不会因为动作设计不合理导致机器人摔倒以及程序不合理导致硬件损坏。

论文通过状态转换模型解决了教育机器人辅导家庭作业中的多轮交互问题和多角色参与问题，并通过实验验证了模型的有效性。下一步工作是在实际的家庭环境中对机器人辅导系统进行测试。在实际使用中测试状态转换模型中的状态以及状态的实现是否可以进一步优化和完善。另一方面，数学是理学科目的基础学科，理学科目的问题和数学问题具有相通性。因此依据数学问题解决模型建立起的状态转换模型是否可以通过适当变形并应用于其他理学科目是一件值得尝试的工作。