

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2023

Personalized ECA Tutoring with Self-Adjusted POMDP Policies and User Clustering

Ashwitha Vichuly Jawahar,
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Jawahar,, Ashwitha Vichuly, "Personalized ECA Tutoring with Self-Adjusted POMDP Policies and User Clustering" (2023). *Electronic Theses and Dissertations*. 9055.
<https://scholar.uwindsor.ca/etd/9055>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Personalized ECA Tutoring with Self-Adjusted POMDP Policies and User Clustering

By

Ashwitha Vichuly Jawahar

A Thesis

Submitted to the Faculty of Graduate Studies

through the School of Computer Science

in Partial Fulfillment of the Requirements for

the Degree of Master of Science

at the University of Windsor

Windsor, Ontario, Canada

2023

© 2023 Ashwitha Vichuly Jawahar

Personalized ECA Tutoring with Self-Adjusted POMDP Policies and User Clustering

By

Ashwitha Vichuly Jawahar

APPROVED BY:

Z. Zhang

Faculty of Education

A. Abusitta

School of Computer Science

X. Yuan, Advisor

School of Computer Science

May 15, 2023

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

An Embodied Conversational Agent (ECA) is an intelligent agent that enables real-time human/computer interaction in natural language. For its rich style of communication, ECA is particularly popular and useful in applications such as education, e-commerce, healthcare, finance, marketing, and business, where a human-like conversation is more attractive to users than traditional keyboard-based interaction. The interest in using ECA in e-learning has become even stronger since the COVID-19 outbreak, and a preliminary investigation has been started by our research group to extend collaborative learning in a virtual environment with personalized ECA tutoring.

This thesis document first highlights the prior work of personalized tutoring with ECA, including wavelet transformation for user clustering and face-to-face interaction for quiz-style e-learning. An enhanced approach is then developed to enable self-adjustment of POMDP policies for dialogue management and to allow a more natural way of question/answer style of personalized tutoring with a generic, flexible tutoring ontology. In addition, the proposed approach uses machine learning techniques to adjust knowledge levels of user clustering and evaluates its effectiveness by conducting experiments with real datasets. This research work is projected to further improve online learning with ECA serving as a personal tutor.

Keywords: User Clustering, POMDP, Personalized Tutoring, Ontology, ECA

ACKNOWLEDGEMENTS

I would like to thank my professor Dr. Xiaobu Yuan from the School of Computer Science for the constant guidance and support for my selected topic and for helping me during the entire year and a half with the efforts to improve on every aspect of the thesis.

I would also like to thank the Faculty of Graduate Studies of the university and my thesis committee members: Dr. Zuo Chen Zhang and Dr. Adel Abusitta for constant support and appreciate the guidance, suggestions, and encouragement along the way. This work extends that of Tristan Szucs and although he has graduated, he was generous enough to help me with the project and was always available in case I ran into blocks or trouble with the project.

I would also want to acknowledge the support and guidance I received from my sister Abhinaya Vichuly Jawahar and my brother-in-law Balasubramaniam Arunachalam. And thank my mother Kasthuri Munirathinam for her constant love and support, without which I could never have moved to a completely other country.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY.....	iii
ABSTRACT.....	iv
ACKNOWLEDGEMENTS.....	v
LIST OF FIGURES.....	ix
LIST OF TABLES.....	xi
1 Introduction.....	1
2 Literature Review	4
2.1 Online Learning	4
2.2 AI in Education	4
2.3 Embodied Conversational Agents	6
2.3.1 Dialogue Management Systems.....	6
2.3.2 POMDP.....	7
2.3.2.1 POMDP With Belief State History	8
2.4 Ontology	9
2.4.1 Steps in building an ontology	10
2.5 User Clustering	12
2.5.1 User clustering in e-learning systems	12
2.5.2 DWT for user clustering	13
2.6 Q Learning for Self-adjusting of Policy and NCPS.....	15

3 Prior Work	20
3.1 AI in education.	20
3.2 POMDP-Based Dialog Management.	22
3.3 Knowledge Level Estimation Using DWT.	24
3.4 Embodied Conversational Agent	25
3.4.1 Lip synchronization and facial expressions	26
3.5 Ontology.	29
4 Problem Statement and Proposed Methodology.	32
4.1 Problem Statement And Contributions	32
4.2 Overall Architecture	33
4.3 Algorithm	35
4.4 Algorithm Details	38
4.4.1 Initialization	38
4.4.2 Knowledge level determination.	38
4.4.3 Ontology flow	38
4.4.4 Q-Learning for Self-Learning of POMDP Policy.	40
4.4.5 Q-Learning for Self-adjusted User Clustering.	44
4.4.6 ECA Simulation.. . . .	46
4.4.7 Ontology Result.. . . .	46
4.5 Running Example:	47
5 Implementation and Experiments	52
5.1 Software and Tools	52
5.2 Dataset	53
5.3 POMDP Policy Selection.	53

5.4 Ontology Structure.	54
5.4.1 Visualization in Protégé:	54
5.5 User clustering.	56
 6 Experimental Results	 58
6.1 User Clustering Results.	58
6.2 Personalized Questions and Hints/Feedback	59
6.3 Success Rates.	62
6.4 Limitations.	64
 7 Conclusion	 66
7.1 Research Summary	66
7.2 Future Work	67
 Appendix A	 68
Useful Links	68
References	69
Vita Auctoris	75

LIST OF FIGURES

Figure 1 POMDP Overview.	8
Figure 2 BSH in POMDP.	9
Figure 3 DWT Wavelet Example.	14
Figure 4 Q Matrix / Table.	16
Figure 5 Q-learning Cycle.	17
Figure 6 AirSim ECA.	26
Figure 7 Fuzzy Rules for Sentiment Analysis.	27
Figure 8 VanLehn’s Schema for Tutoring Ontology.	29
Figure 9 Overall Architecture.	34
Figure 10 Dialog Actions.	39
Figure 11 Belief State History Example.	42
Figure 12 Transformed DWT Example.	43
Figure 13 OWL Class Structure.	55
Figure 14 OWL Properties.	56
Figure 15 Personalized Questions Results.	60
Figure 16 Success Rate Result.	63
Figure 17 Efficiency Comparison.	64

LIST OF TABLES

Table 1 Q - learning Initial Example.	18
Table 2 Q values Updated.	19
Table 3 Prior Work on AI in Education Summary.	21
Table 4 Prior Work on POMDP Summary.	24
Table 5 Knowledge Level Boundaries.	25
Table 6 Prior Work on Ontology Summary.	30
Table 7 Algorithm.	35
Table 8 Example Conversation History.	40
Table 9 Self-learning of Policy Example.	41
Table 10 Self-learning of Policy Updated Q table.	44
Table 11 Q table for Self-adjusted User Clustering.	42
Table 12 Updated q table for Self-adjusted User Clustering.	45
Table 13 Sample Interaction Example 1.	48
Table 14 Sample Interaction: Example 2.	49
Table 15 Sample interaction for Example 3.	51
Table 16 Software And Tools	52
Table 17 Components of Datasets.	53
Table 18 Instance 1: Geography.	55
Table 19 User Clustering	59
Table 20 Summary of Personalized Questions and Hints.	61

Chapter 1

Introduction

Online learning or e-Learning has roots that are firmly planted in the education sector. With the increasing speed of internet connections, opportunities for digital training have arisen. Since the internet and education have combined to provide individuals with the chance to learn new skills, there has been a huge increase in online learning during the past ten years [10]. The global pandemic has significantly altered how students study globally, resulting in distinctive online learning. Worldwide students have abruptly switched from classroom instruction to online instruction. Researchers have predicted the market for online education to reach \$350 billion by 2025 even before the pandemic [10]. Therefore, the numbers may have changed after examining the growing effects of COVID-19 on the market. Online learning firms contain a vast quantity of user data, allowing such platforms to utilize machine learning algorithms that can improve students' learning habits. Generally, pattern recognition is used in machine learning algorithms, which allows for the personalization of material for each user. For instance, the platform can change the e-learning content to give more in-depth information to assist a student who struggles with a subject during the course. Online learning can occasionally pose hurdles despite these benefits. Some of them include slow internet, download difficulties, audio/video issues, inadequate course content, lack of motivation in the absence of a real teacher, etc. [1]

Since the COVID-19 outbreak, interest in adopting Embodied conversation agents (ECAs) in e-learning has grown even more, and our research team has begun a preliminary study to enhance collaborative learning in a virtual environment with personalized ECA tutoring. In modern education, personalized learning has become

a key focus. Students will acquire more knowledge about the subject if the teaching methods are personalized to their needs, interests, and ability [34]. ECAs encourage learner motivation, participation, and self-assurance and may assist in preventing and treating negative emotional states, such as disappointment and fear of failure, in students [48]. The usage of AI (artificial intelligence) in education is not a choice; rather, it is a shift in education [9]. When AI is used in AI-powered education, it provides the possibility for “more personalized, flexible, inclusive, and engaging” learning and a more sophisticated learning environment for students [29]. With the help of conversational AI, educational institutions can give students personalized experiences based on their interests, level of knowledge, and individual needs.

ECAs are animated anthropomorphic interface agents that can engage a user in real-time, multimodal dialogue while imitating human face-to-face contact using voice, gesture, gaze, posture, accent, and other verbal and nonverbal behaviors [54]. ECAs are frequently employed in social dialogue systems, which function better when given emotional and facial signals. Also, there are several applications such as clinical psychology, e-commerce platforms, real estate sales, etc. where research is done on the effectiveness of ECAs. It has been demonstrated that ECAs increase learner motivation and help students recover from negative states like dissatisfaction and failure fear [54].

The aim of this thesis study is to build on the earlier work of our research group. The ECA created in [11] makes use of a POMDP-based dialog manager to function as an SDS (Spoken Dialog System). To assess the user's knowledge level, discrete wavelet transformation is used on the history of belief states.

To make the dialog system as realistic as possible, the earlier work also used lip-synchronization and sentiment analysis for the avatar. This thesis research aims to use Dialog Management and knowledge selection to provide a personalized learning experience for the end user by merging the prior system architecture with a scalable

tutoring ontology. An enhanced approach is proposed to enable self-adjustment of POMDP policies for dialogue management and to allow a more natural way of question/answer style of personalized tutoring with a generic, flexible tutoring ontology. In addition, the proposed approach uses machine learning techniques to adjust knowledge levels of user clustering.

The rest of this thesis is structured as follows: Chapter 2 summarizes the literature reviewed to help grasp the ideas and provide evidence for the thesis's problem statement. It will discuss if using an ECA as a personalized tutor is feasible. The research team's earlier work, which served as the foundation for this thesis, is reviewed in Chapter 3. The problem that this thesis seeks to answer and the approach that will be taken are both covered in detail in Chapter 4. Chapter 5 will describe the project's overall architecture, the algorithm the code uses to arrive at the desired outcome, and how my contribution fits into it. In order to explain how the suggested system functions, Chapter 6 presents implementation results. Chapter 7 analyzes the results from the previous stage and Chapter 8, discusses the final remarks.

Chapter 2

Literature Review

This chapter provides a summary of the literature about online learning and how AI in education might improve it. The use of Embodied Conversation Agents in the field of education is also examined, as is the importance of creating an appropriate ontology for a particular domain.

2.1 Online Learning

Online learning or e-learning is frequent terminology for web-based learning. In essence, it entails studying through online classes. Via the Internet, it is possible to conduct video conferences, live seminars, and email communication. E-learning is "the delivery of training and education via networked interaction and a range of other knowledge collecting and dissemination technologies," according to experts in education and educational technology [53]. Through e-learning, teachers and students may collaborate in person while leveraging technological resources to improve the learning process [17].

2.2 AI in Education

The usage of AI in education is not a choice; rather, it is a shift in comparative education [8]. AI has previously been used in education, especially in various tools and assessment platforms that aid in skill development. The goal is that as AI educational solutions continue to develop, they will help close gaps in learning and teaching and free up schools and teachers to accomplish more than before. The ideal

use of AI in education is one in which teachers and machines collaborate to get the best results for students.

As explained in [2], some of the benefits of applying AI in everyday learning are:

- Students can enhance their personal learning through artificial intelligence.
- Artificial intelligence is a tool that educators may use to support focused instruction.
- Artificial intelligence can aid in making more accurate decisions for educational administrators.
- It aids in accelerating students' individual learning and success rates.

However, it is also a challenging task to introduce AI in education. One of the most faced challenges is incomplete algorithm design. As the algorithms used in artificial intelligence in education are typically created by outside organizations, they could not meet the needs of the users who are at the forefront of educational technology. As a result, artificial intelligence advances in education frequently contain unmet promises and fail to recognize the needs of their target audiences. Another challenge is that artificial intelligence applications rely on a variety of data sets, some of which may include sensitive data like identity numbers, student photographs, addresses, etc.

2.3 Embodied Conversational Agents

Intelligent virtual agents (IVAs), commonly referred to as conversational agents (CAs), are computer programs designed for natural conversation with human users. A conversational agent is called an "embodied conversational agent" (ECA) when the agent is animated with a visual representation (face or body) on-screen. When compared to non-embodied CAs, ECAs may have an advantage since they may enhance their communication using nonverbal signals like body language and facial expressions. The techniques and tools required for the ECA's operation are covered in this section.

2.3.1 Dialogue Management Systems

Most applications that use ECAs for human-computer interaction utilize dialog managers (DM) to determine responses in reacting to user inputs. For dialog management, a wide range of options is available, including MDP (Markov's Decision Process), FSM (Finite State Machine), a frame-based approach, etc.

The finite state approach uses a finite state machine to model the conversation, including transition, and it has the limitation that the system should be entirely pre-modelled. The frame base approach is more flexible, as this type of DM knows what information it needs, and it asks questions to elicit this information [49]. Its drawbacks include the frame's ability to handle only basic information points.

Like the frame-based method, the information state approach can also contain extra data points about mental states, goals, and other important data that is not directly examined [49]. Here, unlike in Partially Observable Markov's Decision Process (POMDP), the state's data points are not probabilistic. Plan-based approaches have a common aim that users must achieve through actions, which include speaking. Due to the difficulty in predicting what the machine would do, this technique has received

harsh criticism [41]. An agent-based approach allows the system to amend its previously known data points since it sees the dialogue as a collaborative activity [41]. However, this method depends on being aware of the user's objective. In the model used by [49], the technique used for dialog management is primarily POMDP. Its operation also uses other techniques such as BSH (Belief State History) and DWT (Discrete Wavelet Transform).

2.3.2 POMDP

This section is an overview of how POMDP functions, the concepts described herein primarily come from [28]. Markov's decision process (MDP), forms the basis of POMDP, which consists of states and actions. States describe the current state that the system is in, and actions are the various things that the system can do. However, unlike typical MDPs, the current state of POMDP is not fully known, and the system relies on observations to determine what the current state is. These observations do not specify an exact state but instead give us a probabilistic estimate on what the state is, from which the system can build a probability distribution over all the states using the available transitions and these observations.

POMDP is a beneficial method for managing "spoken dialog." It can be described as a 7 tuple (S, A, T, R, Ω , O, Y) such as states, actions, transitions, rewards, observations, conditional observation probabilities, and reward discount.

- State: various states the POMDP could believe it is in at any moment.
- Action: available actions the POMDP has available to take.
- Transitions: different ways in which the states can transit from one state to the next.
- Rewards: immediate reward for making specific transitions.
- Observations: observations the POMDP can observe to decide its belief state.

- Conditional observational: probabilities represent how each observation affects the belief state.
- Reward discount: factors the POMDP system uses to prioritize immediate or future rewards.

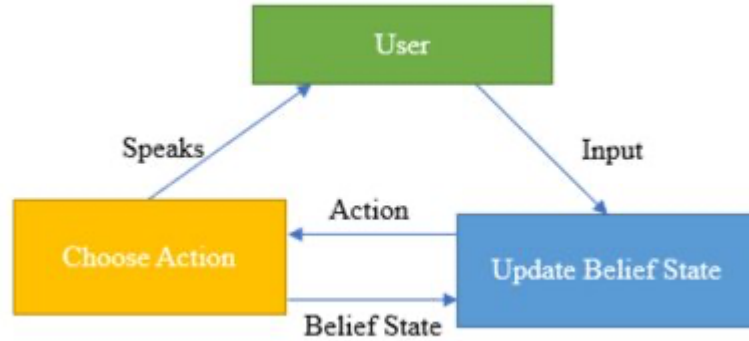


Figure 1 POMDP Overview

Figure 1, illustrates how the POMDP works, emphasizing the dialogue loop. After the system performs an action, the user provides input, which is the observation from the perspective of the system. Inside the system, it uses this input to update the belief state. It passes the last action back to the belief state calculator as it allows for the observations to be taken in context as they are a reply to the last action the system used. This belief state determines the next action taken.

2.3.2.1 POMDP With Belief State History

A belief state is referred to as a probabilistic estimate of what the state is, from which the system may create a probability distribution across all the states using the transitions and observations that are currently available as shown in Figure 1. Keeping track of history has been proven to be beneficial. This is referred to as Belief State History (BSH) [40], and this history may be examined to provide further

information about the user. In order to obtain an estimate of the state, the system must identify the probability state pairings that make up the belief state.

Figure 2 demonstrates the POMDP interaction between the user and agent [40]. In a POMDP, we add a set of observations to the model. Therefore, the state provides us with an observation that offers us a suggestion about what condition it is in rather than simply observing the present state. As a result, we also must specify an observation function because the observations may be probabilistic. The probability of each observation for each state in the model is simply provided by this observation function.

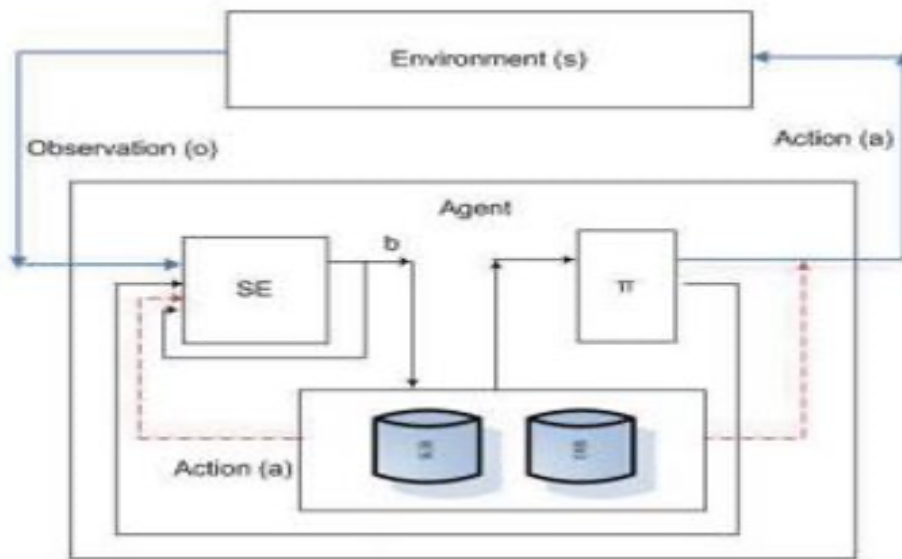


Figure 2 BSH in POMDP [40]

2.4 Ontology

Ontology is a formal, explicit specification of a shared conceptualization [52]. For researchers who need to share information within a domain, an ontology defines a common vocabulary. It includes machine-interpretable definitions of the

fundamental concepts in the domain and the relations between them as well [52]. Two methods have been developed to create a tutoring ontology. The method that uses formal language to express knowledge comes first. Ontology Web Language (OWL) and other formal ontology frameworks are part of this. The second method is based on data mining and machine learning. This methodology comprises models that make use of pattern mining, artificial neural networks, etc.

Ontology describes concepts, their properties, and limitations on how they may be used in a domain. Ontologies enable clear and exact access to structured knowledge while allowing enough freedom for the development of the final software [18]. In e-learning, ontology can help improve the definition of the domain of course knowledge, in the assessment phase and in the generation of an adapted path of learning.

The author of the paper [52] lists a few important factors to consider while creating an ontology. The process of creating ontologies is dynamic and iterative. Several techniques may be used to model ontologies. An ontology's classes and characteristics should closely resemble real-world objects.

2.4.1 Steps in Building an Ontology

A concise step-by-step instruction is provided in [52] and also described below for building an ontology using OWL.

1. Determining the domain and scope of the ontology: This step includes asking important questions to reduce backtracking and redoing the ontology. Some basic questions to ask before starting to build an ontology are "What is the domain that the ontology will cover?", "Who will be the user of the ontology" and "What questions

will the ontology answer?". The answers to these questions might change with iterations but it helps to limit the scope of the ontology.

2. Consider re-using existing ontologies: As creating an ontology is a time-consuming and intricate process, the author advises studying existing ontologies and using all or some of their components rather than starting from scratch to save time and resources. Ontolingua, the DAML Ontology Library, and others are a few examples of such ready-to-use ontology libraries [53].

3. Enumerate important terms in the ontology: This mostly involves making a list of the terms and properties that we want to include in the ontology. Even if it is not specific and even if some of the properties overlap, a detailed list is quite helpful.

4. Define classes and class hierarchy: Since the next and current steps are interdependent, both must be completed at the same time. According to the literature, one of the following methods can be used to construct class hierarchies: Bottom-Up approach, Top-Down approach, and Hybrid approach. The methodology is solely reliant on the developer's perception of the classes and comprehension of the domain. They are all equally as good as one another.

5. Define properties of classes: Typically, defining the classes by themselves is insufficient for an ontology to work. To specify the internal structure, the developer must define the properties of the classes. The wine's color, such as red, white, or rose, as well as its body, flavors, and sugar content are examples of potential properties. Traditionally, every property corresponds to a class. The class "Winery", for instance, will have a property called "location", while the class "Wine" can have properties like "color" and "taste."

6. Define facets of the slots: The domain and range of a property are examples of facets of the slots. For illustration, the term "produces" can be used to describe regions' class values which in turn yield wines' class values. Individuals belonging to the class "Region" make up the domain of the property "produces" in the example, whereas individuals belonging to the class "Wine" make up its range.

7. Create instances: This is the final stage in creating an ontology's early draft. The unique values in each class are known as instances. The following is an illustration of an instance of the class that describes a particular sort of wine and includes attributes like Body, Color, etc.

Body: Dark

Color: Blue

Flavor: Delicate

Grape: Gamay (instance of the Wine grape class)

2.5 User Clustering

User clustering is the process of identifying groups of user data based on some similarity measure [46]. Hierarchical and partitional clustering approaches are two well-liked user clustering methods. The data can be divided into a cluster tree using hierarchical approaches, while parallel clusters are created using partitional techniques.

2.5.1 User Clustering in e-learning systems

The system must evaluate users' knowledge levels, preferences, and learning styles to create a personalized experience for online learning. To satisfy the needs of

individual students, a personalized e-learning system is needed. Even among students who are enrolled in the same course and have comparable technical skills, the need typically varies [21]. The order in which the material is organized and how it is presented may have a significant impact on how actively a student engages with a learning system [21].

2.5.2 DWT for user clustering

One approach for assessing multiple signal types is the wavelet transform (WT). A more efficient way to encode a signal's time and frequency is via the discrete wavelet transform (DWT), a specific instance of the wavelet transform [27]. The sampling of WT occurs in discrete time for discrete datasets. The equation for DWT is given below:

$$W_q(a, t) = \frac{1}{\sqrt{a}} \sum_{n=1}^N q(n) \varphi\left(\frac{n-t}{a}\right)$$

In the above equation,

n - current data that is being analyzed, which has size N.

φ - mother wavelet

t - discrete time

a - scale

Examples of how DWT transforms a wave and obtains its variation points are shown in Figure 3. In the figure, the trend varies depending on the windows utilized as shown by the asterisk on the top graphs. The locations where the new wave crosses zero on the lower graph are shown with an asterisk, indicating that they signify where

the trend is changing. Due to the bigger window in the left window, fewer variation points are found there.

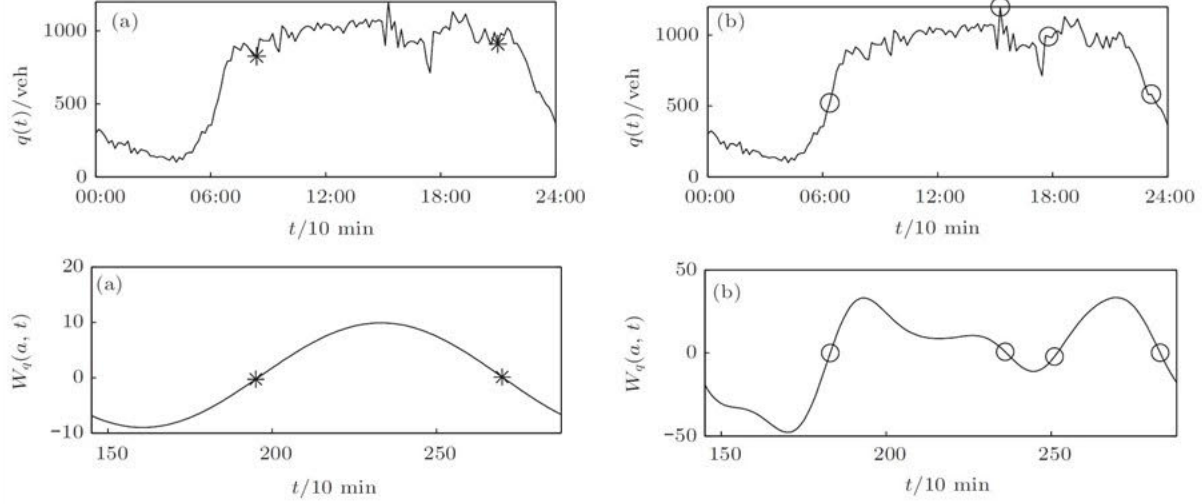


Figure 3 DWT Wavelet Example [27]

The DWT method is frequently used to evaluate non-stationary signals, such as audio. When DWT is applied to a wave, the wave exhibits several sharp variation points [11]. This unique property of DWT may be utilized to identify a student's consistency of performance in a personalized tutoring model. The concept behind utilizing DWT for user clustering is that for an experienced user, this variation will be relatively stable as these variation points identify the position where the trend of belief states has changed. Depending on the number of change points, users may be divided into different knowledge levels [24].

For the purposes of this thesis, the Discrete Wavelet Transformation method employed in [24] is utilized to categorize system users according to their knowledge level.

2.6 Q Learning for Self-adjusting of Policy and NCPS

Machine learning (ML) methods use instances to train the system how to predict unknown values. The ML technique known as reinforcement learning (RL) allows the system to learn from each action taken in an unknown space, observing a reward in order to enhance future behavior [55].

Q-learning is a model-free reinforcement learning method that learns the significance of a certain action in each state and uses q values to predict the best behavior.

Q-Learning consists of states and actions. A state is the system's present state. An action, on the other hand, is one of the various steps the system may take. It is important to note that not every state can be converted into an action, thus an action is not just a clear transition. An action does not generate a known state, but rather an unknown state. All potential current state and action combinations are represented in the matrix Q. The values in Q matrix are known as "q-values" and they reflect the possible reward from doing this action. The system receives a reward value r, or the immediate reward, immediately after performing an action. This reward and the subsequent best q-value are used to modify the initial q-value. The q-value used to choose an action should only be changed after the action has already been completed. An example of a q table/matrix is shown in Figure 4.

The equation for adjusting q- values is shown below.

$$Q(x, a) = (1 - \alpha) Q(x, a) + \alpha (r + \gamma \max_b (Q(x, b)))$$

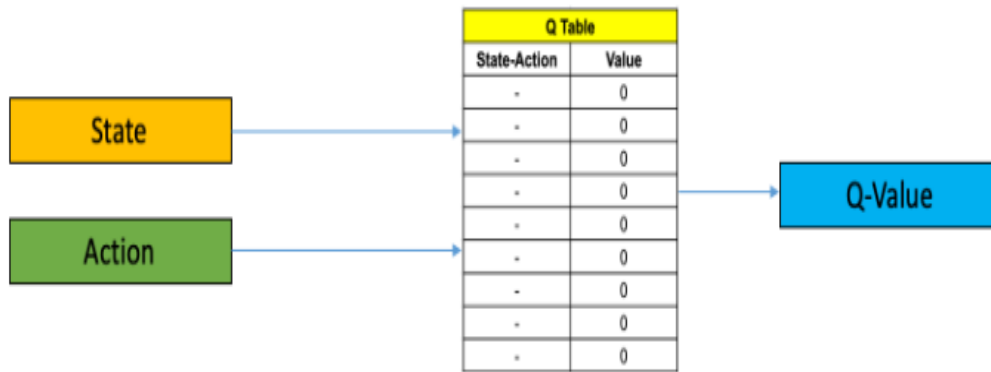


Figure 4 Q Matrix / Table

In the above equation,

α - is the learning rate

X - is the current state

a - is the last taken action

Y - is the new state

b - is all actions

γ - is the reward discount

In the first part of the equation, which is the left-hand side of the major addition, tells how much of the new value was applied to the previous value. The system does not learn exclusively from the new action but also considers past actions, as shown by the example, where 20% of the old value is maintained if the learning rate is 0.8. The remaining 80% of the new value is calculated in the second half, which also reflects the value learnt as a result of the previous action. The new component takes the reward that was immediately returned by the previous action and adds a value to reflect the reward that will be given in the future based on the changed state that was noticed after the action. This potential reward has the highest q-value in the recently discovered state b . This future reward is restrained by the reward discount, which establishes the relative importance of the present reward and the future reward.

As shown in Figure 5, the flow of Q-learning cycle algorithm goes through the following steps.

- The system executes the action with the highest q-value based on the q-values of the current state.
- The system now receives a reward value of r and is in a different state, or it might be in the same state.
- Based on the new state, the optimum course of action, and the reward obtained, modify the q-value that was previously used.

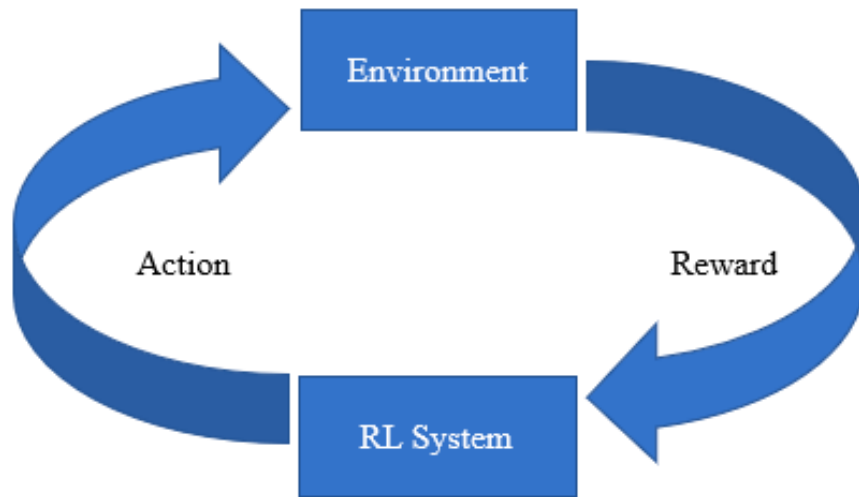


Figure 5 Q-learning Cycle

Table 1 illustrates the q table for a system that consists of four states and four actions. This example will use a reward discount of 0.5 and a learning rate of 0.8.

State	Action 1	Action 2	Action 3	Action 4
0	5	9	3	1
1	8	3	5	9

2	6	9	2	4
3	4	5	7	9
4	3	4	5	4

Table 1 Q - learning Initial Example

If the system's current state is 2, from the q table the best action would be action 2 as it has the q value "9" which is greater than the other actions for that state. After taking action 2, the system transits to state 3 and receives a reward of 3. Based on this data, the system updates the q-value for state 2 action 2 using Eq.

$$Q(2,2) = (1 - 0.8)Q(2,2) + 0.8(3 + 0.5 \max_b(Q(3,b)))$$

Similarly, by choosing the maximum q value, action 4 is the optimum option for state 3.

$$Q(2,2) = (0.2)(9) + 0.8(3 + 0.5(9))$$

$$Q(2,2) = 5.2$$

The modified state 2 action 2 is shown in Table 2, which is the new Q-Value table:

State	Action 1	Action 2	Action 3	Action 4
0	5	9	3	1
1	8	3	5	9
2	6	5.2	2	4
3	4	5	7	9

4	3	4	5	4
---	---	---	---	---

Table 2 q values updated.

Until the system reaches an end state and closes, this procedure is repeated. In q-learning, the final state is decided externally rather than being stated.

Chapter 3

Prior Work

This section provides a summary of the key concepts and methodologies that this research expands upon. This thesis draws mainly from the prior work of our research group presented in [1] and [11].

3.1 AI in Education

The use of artificial intelligence in education ranges from conversational agents for personalized tutoring, analysis of student writing, intelligent agents in game-based environments, chatbots for student support, and student/tutor matching that firmly places students in control of their own learning. The obstacles or misperceptions around AI in education have come to light, as a result of the interview and the review study [35] [37]. A comprehensive assessment standard must be developed to assess AI's efficiency in education. According to Woolf's [42] Roadmap for Education Technology, in the era of AI Educational Data Mining, it will be possible to measure the success and failure of teaching methodologies as well as the lifetime evaluation of student's knowledge, progress, and learning environments.

Table 3 provides a summary of the recent works that predominantly support this thesis.

Author(s)	Title	Contribution / Key Points
Hyangeun Ji, Insook Han & Yujung Ko	A systematic review of conversational AI in language education: focusing on the	It suggests guidelines and recommendations for teachers and AI researchers.

(2022) [2]	collaboration with human teachers.	It also talks about roles of conversational AI and teachers in learning.
Hwang, S (2022) [5]	Examining the Effects of Artificial Intelligence on Elementary Students' Mathematics Achievement: A Meta-Analysis.	This study examines the overall effectiveness of AI on elementary students' mathematics achievement using a meta-analysis method.
Yang, H., & Kyun, S. (2022)[3]	The current research trend of Artificial Intelligence in language learning: a systematic empirical literature review from an activity theory perspective	This study uses analytical framework for analyzing the need, activity and outcome of the technology–supported learning environment.
Shahriar, Asif & Weber, Peter (2022) [4]	Talk the Talk: Enhancing the Educational Chatbot Experience by Conversational Design.	It uses conversation design to improve the naturalness and flexibility of our multi-entity chatbot.

Table 3 Prior Work on AI in Education Summary

3.2 POMDP-Based Dialog Management

As part of the on-going project carried out by our research group, this study relies on the work completed by several former students. Tristan's work was followed by a series of papers that made improvements to the software avatar's dialogue management. The main objective of Tristan's study [11] was to enhance system intuition by merging the POMDP model with Belief State History (BSH) and COCOM for policy selection, even though other works have contributed to the demonstration of the effectiveness of a probabilistic model for dialogue management. Previously, Sathulla [42] incorporated COCOM into the model. The major obstacle this feature overcomes is related to the problems with dynamically addressing POMDP policies.

Moreover, Bian introduced the usage of Belief State History with a POMDP-based model in [40] in 2010, which allowed the model to consider both the previous state and the present state when choosing the next action to take. The redesigned model also employs ontology as the domain knowledge for making decisions. The next action is preserved as the final one if it is successful; else, a new corrector action is chosen. Domain knowledge was necessary for this adjustment, as it is the source of knowledge for any tutoring system. Mulpuri in 2016 [27] offered the concept of utilizing Belief State History (BSH) to compute the number of Change Points (NCPs), which is used to evaluate the knowledge level of the user. Both improvements mentioned above were included in the model of this study. One significant departure from Bian's work that can be noticed in Mulpuri's work is that the system analyses the trend in the complete BSH to predict potential changes rather than going back one step to decide the next action. As a result, the POMDP technique may use the entire history to detect long-term changes in the subsequent set of actions.

Similar reasoning is used with reinforcement learning (RL) in Tristan's investigation. The q-learning model uses reinforcement learning to update the system's most recent mode whenever the system obtains a reward for entering the subsequent state. The dialogue management system considers this updated mode when determining the next action to take. The following is the updated q-learning equation for each mode, where Q' is the new mode in the equation.

$$Q(x, a) = (1 - \alpha) Q(x, a) + \alpha(r + \gamma \max_b (Q'(y, b)))$$

Although the models used in other research on POMDP differ from those in this thesis, they are nevertheless useful for learning about the method's practical applicability. In the research described in [33], a Gaussian process with POMDP dialogue management is introduced, which automatically updates POMDP policies and removes the need for a manual generation. In order to identify the user's knowledge level for the purposes of this thesis, the Discrete Wavelet Transform is utilized in association with the POMDP dialogue manager in the works of [11], [27], [33].

Table 4 provides a summary of the works that predominantly support this thesis.

Author(s)	Title	Contribution/Key Points
Vijaya Krishna Mulpuri (2016) [27]	Trend Analysis of Belief-State History with Discrete Wavelet Transform for Improved Intention Discovery	Modelled the collection and usage of POMDP's BSH. Demonstrated the usage of DWT on BSH to obtain the NCP of BSH. Created several POMDP policies based on COCOM for various knowledge levels.

Ruturaj Rajendrakumar RAVAL (2019) [16]	An Improved Approach of Intention Discovery with Machine Learning for POMDP-based Dialogue Management	RL policies were suggested to replace hand-crafted policies with self-adjusting policies, but no implementation was suggested.
Tristan Szucs (2020) [11]	Lip Synchronization for ECA Rendering with Self-Adjusted POMDP Policies	Demonstrated how the system takes optimal action by using Self-adjusting POMDP policies (RL techniques). Also incorporated a method to match lips to the generated ECA audio and the automatically chosen emotion.

Table 4 Prior work on POMDP Summary

3.3 Knowledge Level Estimation Using DWT

POMDP has been widely employed for dialogue management in communicative models, as highlighted by [11]. However, most of them just consider the present state when deciding what to do next. A time-data series cannot be created or observed for patterns using other approaches like sampling and histograms. Our dialogue management system maintains a Belief State History during the transaction, thus bringing about the advantage of using DWT for data analysis [27]. The capability of DWT to eliminate data noise also helps to produce more accurate results. Applying DWT to the wave of BSH produces a number of sharp variation points (NCP), each of which signals a sharp fluctuation and results in a change of trend in belief states.

A variety of NCPs may be used as a foundation for classifying users' knowledge, and this information has been used in [1] to categorize users according to their level of knowledge when the NCP is in a particular range as shown in Table 5. For example, the user is considered a beginner if the NCP is between 7 and 9 inclusively.

NCP	Knowledge Level
NCP<4	Expert
NCP<7	Professional
NCP<10	Beginner
Otherwise	Novice

Table 5 Knowledge level boundaries

This thesis aims to personalize each student's learning based on their current level of understanding. Self-adjustment of the NCP is an important contribution of this thesis to cluster users based on their knowledge level.

3.4 Embodied Conversational Agent

Embodied Conversational Agents (ECAs) are intelligent avatars that can converse with humans and frequently display some form of social behavior, such as altering their facial expressions [49]. Human-like dialogue is more engaging to users than conventional keyboard-based communication in applications including education, e-commerce, healthcare, finance, marketing, and business. ECA is well-liked and beneficial because of its extensive communication style.

The software tool AirSim is used in Raturaj's work [16], which was extended by Tristan in [11], to create a software avatar that serves as an ECA. The dialogue agent that serves as the personal instructor in this thesis is an expansion of the work from

[16] and [11]. Figure 6 illustrates the developed ECA with lip synchronization in mutual facial expression.



Figure 6 AirSim ECA

3.4.1 Lip Synchronization and facial expressions

Tristan's main contribution to the project is accurately synchronizing the software avatar's lips in order to make it appear more human. Here is a list of the processes that were approximately conducted in that order before the ECA's final visualization:

- **Sentiment Analysis:** The system employs fuzzy logic to decide what emotion the ECA should be exhibiting for the current transaction based on the NCP, reward, and applicable rules. Based on the fuzzy rules shown in the Figure 7, an emotion is chosen.

	Policy			
Sentiment	Strategic	Tactical	Opportunistic	Scrambled
Negative	Disgust	Anger		Fear
Neutral	Fear	Sad	Surprise	Sad
Positive	Happy		Surprise	

Figure 7 Fuzzy Rules for Sentiment Analysis [1]

- **Text-to-speech:** Voicery's text-to-speech module was used in [11] while Google Text-to-Speech was utilized in [1].
- **Phonetic Alignment:** The Pocket- Sphinx speech recognition engine is used to create the forced phonetic alignment module. The text-to-speech engine sends the output text to be spoken together with the voice file. According to the input audio sample, it returns a schedule of the phonetic syllable the ECA should emulate at each point in time. For example, the following phonetic alignment schedule is created for the phrase "Can you please rephrase your request":

[('SIL',0,0,66), ('T',0,67,72), ('F',0,73,80), ('DH',0,81,84), ('UH',0,85,91), ('R',0,92,101), ('UW',0,102,113), ('ER',0,114,117), ('B',0,118,130), ('IH',0,131,133), ('F',0,134,137), ('W',0,138,146), ('ER',0,147,151), ('UW',0,152,169), ('S',0,170,184), ('G',0,185,191), ('W',0,192,200), ('AA',0,201,207), ('F',0,208,231), ('AA',0,232,243), ('UW',0,244,252), ('UW',0,253,260), ('S',0,261,274), ('HH',0,275,278), ('UW',0,279,287), ('AO',0,288,307), ('AA',0,308,312), ('B',0,313,322), ('OW',0,323,326),

('TH',0,327,334), ('N',0,335,338), ('AW',0,339,345), ('ER',0,346,362),
('OW',0,363,367), ('AH',0,368,370), ('S',0,371,396), ('P',0,397,401),
('S',0,402,410), ('P',0,411,418), ('SIL',0,419,466), ('TH',0,467,477)]

In the example, the alignment module first sets all Action Units (AU) associated with the facial unit to zero in order to calculate the face. Now suppose it is at the related point of 195; the current phenome is the 'w' phenome, which relates to the phenome f AU. This sets the 'f' AU intensity to 0.5. Before that in the alignment is the 'g' sound (phenome g AU) and therefore is an intensity of 0.1. These two are combined with the raised cheeks and sent to the ECA for rendering. The audio will play, and the system will loop back to the start. Finally, at the end of the audio file, the system takes the initial happy action units and sends them to the ECA again to render.

- **Result:** As a result, the ECA utilizes the audio file (a text-to-speech module output), sentiment (a sentiment analyzer output transformed to a text-to-speech engine-supported emotion), and lip schedule (a PocketSphinx module output) to speak at the same moment, displaying the appropriate facial gestures and moving the lips in line with the phonetic alignment.

3.5 Ontology

Among the former students involved in the project, Niyati's major contribution was the construction of an ontology for an intelligent tutoring system [1]. Section 2.4.1 discussed the steps for building an ontology, which includes choosing the ontology's domain and scope, thinking about utilizing current ontologies, listing the key concepts in the ontology, class definitions and class hierarchies, defining class properties (Slots), defining the slots' facets, create an instance.

Ontologies have gained immense importance in the field of education. With a wide range of applications from an explicit representation of domain knowledge to automatic content generation based on the user, research and studies on education ontologies are constantly growing. [34]. VanLehn's schema, introduced by Kurt Vanlehn in [47], is one of the most often used structures for a tutoring ontology and is shown in Figure 8. The schema states that an ITS' (intelligent tutoring system) behavior consists of two loops: the outer loop and the inner loop. The outer loop consists of a series of tasks, with an increasing level of difficulty. Each assignment will be harder than the one before it.

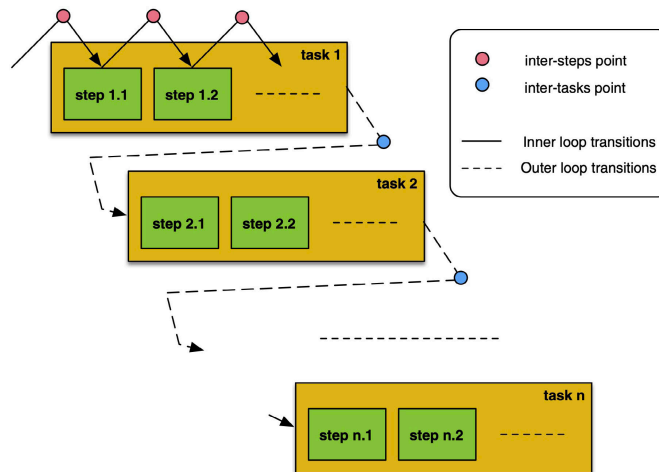


Figure 8 VanLehn's Schema for tutoring ontology

There is an inner loop within each task. This is the progression of smaller tasks that the user must successfully accomplish in order to finish the mission. To assist the learner transition from one inner step to the next at the transition locations shown in Figure 8, the ITS may provide help in forms such as feedback or hints.

The most significant studies from this section that support this thesis are summarized in Table 6.

Author(s)	Title	Contribution/Key Points
Guoying Liu (2012) [36]	A task ontology model for domain independent dialogue management	Developed a task ontology paradigm for managing discourse across domains to achieve domain independent conversation management, ontology and task model will be applied to the dialogue system architecture.
Shubhrendu Tripathi (2016) [30]	A Run-Time Approach of Combining Ontologies to Enhance Interactive Requirements Elicitation for Software Customization	The ontology combination approach has been presented. The developed algorithm dynamically improves the interactive Requirement Elicitation process by combining ontologies at run-time
Niyati Vyas (2022) [1]	An Approach of Using Embodied Conversational Agent for Personalized Tutoring	Constructed a generic, scalable tutoring ontology. Used the ECA dialog management to monitor the progression of belief states and evaluate user knowledge for

		the purpose of adjusting feedback and task offering.
--	--	---

Table 6 Prior Work on Ontology

Chapter 4

Problem Statement and Proposed Methodology

The problem statement is presented in this chapter, along with a focus on the contributions this thesis study has made to the problem under investigation. To meet the objective of the thesis study, the main algorithm of the current system is either expanded or modified. The details of these adjustments and a description of how they function will be covered in the chapter's last section.

4.1 Problem Statement and Contributions

This thesis explores the application of Embodied Conversational Agent (ECA) for personalized tutoring, while numerous researchers are currently working on developing strategies for linear and group learning.

There are two primary objectives of this thesis which aim to advance personalized tutoring research in ECA. This research first describes prior works of personalized tutoring using ECA, including the use of wavelet transformation for user clustering and face-to-face interaction for quiz-style e-learning [1]. A more advanced method is suggested to allow POMDP policies for dialogue management to self-adjust, and to provide personalized tutoring in the form of a more natural question-and-answer format with a generic, adaptable tutoring ontology.

Another significant contribution of this thesis is that the suggested methodology employs machine learning techniques to adjust the knowledge levels for user clustering and evaluates the effectiveness of these techniques through experiments with real datasets.

4.2 Overall Architecture

The overall architecture design is shown in Figure 9. It allows one to observe how information moves from one component to another. Figure 9 and architecture are modified versions of those in [11].

The system begins with the user input (Observation). This user input is passed to the Dialog Management Module. The same input is also passed to the Sentiment Analysis block.

This database contains the ontology which is solely responsible for the storage of domain knowledge. The ontology is responsible for ascertaining the response to be given back to the user, and the steps to be taken by the system in order to proceed. The state estimator determines the current state that the system is believed to be in. The current state is passed to the BSH database in order to maintain track of the history of belief states, and the state estimator creates both the reward needed for q-learning as well as the new belief state. The knowledge level estimator receives the BSH from the storage and uses it to perform DWT on the data to determine the number of sharp variation points. These are summed together to give the system's total NCP count. The system determines the user's knowledge level based on this number. Depending on the user's level of knowledge, the system will choose the new policy to apply within the policy selector.

The state estimator trains the most recent policy and gets rewarded. This module outputs the mode and the recommended actions text for usage outside the POMDP system as its final output.

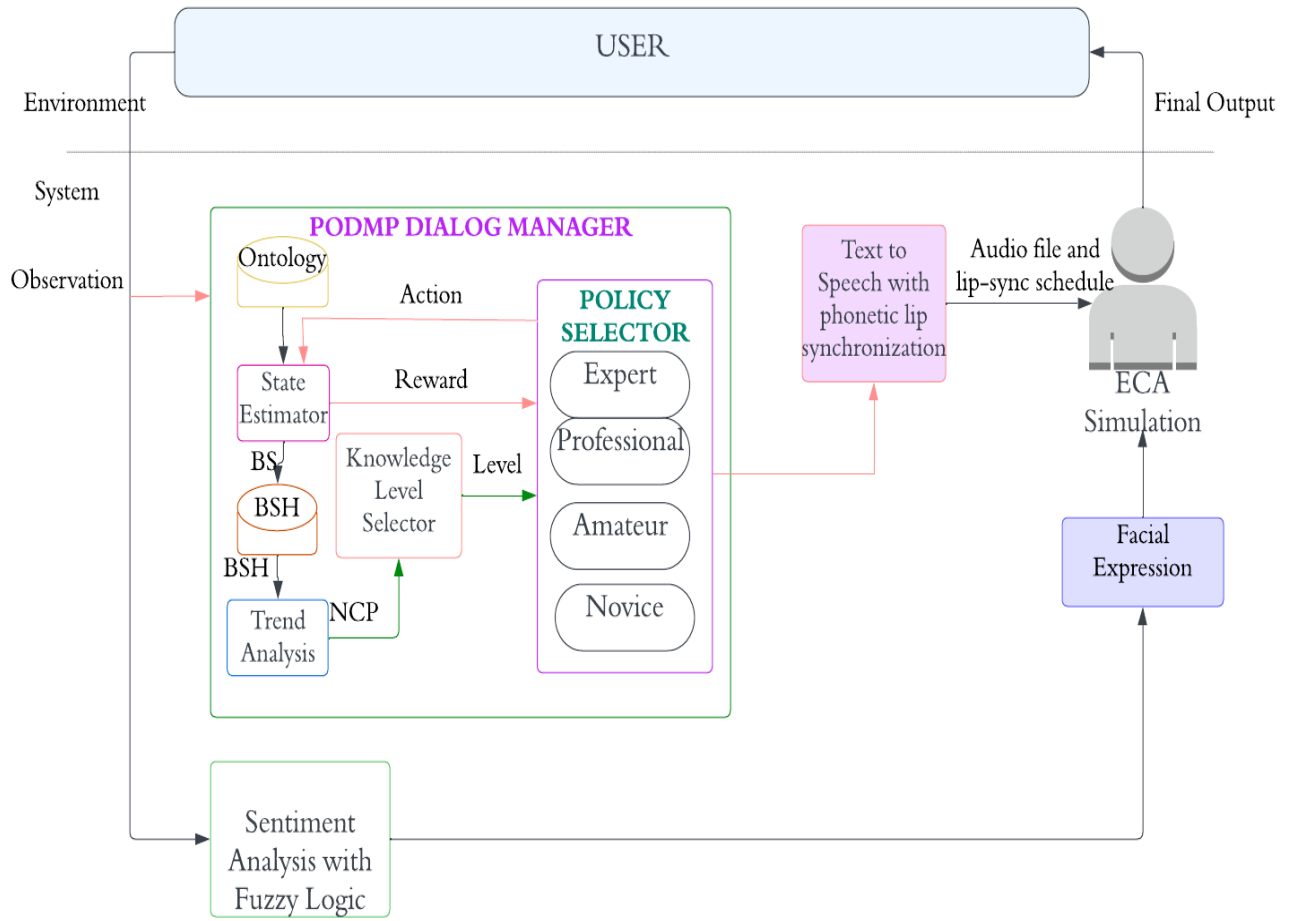


Figure 9 Overall Architecture

To determine the sentiment of user input, the sentiment analyzer employs fuzzy logic. The fuzzifier analyses the user's observation to determine what emotion the ECA might exhibit. The audio is generated by the text-to-speech system by combining the text and the emotion. To obtain the lip sync schedule, the audio and text are merged. The ECA simulation is then generated using the lip sync schedule, audio, and text. The ECA simulation is then generated using the lip sync schedule, audio, and text. The user then watches the ECA and bases its subsequent input on it, ending the conversational loop.

4.3 Algorithm

The algorithm for the architecture of Figure 9 is described in Table 7. The table's highlighted sections signify places where there has been a contribution or enhancement introduced by this thesis research.

Algorithm	
Initialization	<pre>1. BSH = [] 2. endState = false; knowledge_level = None; last_mode = None last_action = None current_task = None current_question = 0</pre>
Select Subject	<pre>Subjects = ['Grammar', 'Math', 'History', 'Geography'] Subject_choice = int(subject_choice)</pre>
Determine knowledge level and Select task	<pre>Knowl = knowledgeLevelSelector(NCP) currentTask ← getTask (selected_subject, task_level, current_task)</pre>
Ontology flow	<pre># Determine level of questions based on user's knowledge level if knowl == 'Beginner': current_task = onto.BeginnerLevel elif knowl == 'Professional': current_task = onto.ProfessionalLevel else: current_task = onto.ExpertLevel</pre>

	<pre> # Get question bundle question_bundle = queBundle(currentQue, currentTask) # Display question and prompt user for answer print(question_bundle['Text'][0]) answer = input() # Check if answer is correct and provide feedback if answer.lower() == question_bundle['Answer'][0].lower(): print("Correct!") if currentQue == 19: print("Congratulations! You have completed the Q&A.") exit() else: currentQue += 1 else: print("Incorrect. Please try again.") action = 'Hint' then hint = getHint(current_question, last_mode) #update the knowledge level of the user </pre>
Calculate belief state and reward	<pre> bel = calculateBelief(isCorrect, last_action) BSH.add(bel) # Add belief to BSH reward = calculateReward(last_belief, bel) </pre>

Calculate NCP(self adjustment)	NCP = forced_NCP knowl = knowledgeLevelSelectorTESTER(NCP, e_lim, p_lim, a_lim) mode = modeSelector(policy_type_to_use, knowl) #self adjustment of NCP range using q learning algorithm
Policy selection	action = mode.getAction(b) #This implementation is q-learning
ECA Simulation	sentiment = getSentiment(text) #Sentiment analysis using fuzzy logic au = getActionUnit(sentiment) eca.setFace # Set Face of ECA according to sentiment audio = textToSpeech(text, voiceaccent) # Google Text to Speech eca.say(audio, audio.text)
Save Interaction	saveInteraction(attemptNumber, currentTask.name, user_answer, "Correct" if is_correct else "Incorrect", question['KCForHint'][0], question['Answer'][0], False, False)
Select Next Task	currentTask = getTask(knowl, False, currentTask)

Table 7 Algorithm

4.4 Algorithm Details

The proposed methodology employs a range of algorithms and several stages to achieve its goal. This section will describe how they work in approximately the same order of the system's operation. A few of the implementation details unique to the experiments will also be elaborated in this section.

4.4.1 Initialization

The simulator will open upon system startup, and the system will start in the empty belief state. The first dialog iteration will start when initialization is completed. To start the first iteration the user will enter in their first piece of dialog by selecting the subject from the given choice.

4.4.2 Knowledge level determination

The user's initial level of knowledge is assumed to be a beginner and the user selects a subject. The user's current question is chosen depending on the chosen subject and knowledge level.

4.4.3 Ontology flow

The user is requested to select a subject before continuing with the task, as shown in the algorithm, once the knowledge level and current task are determined. The system begins a loop through every question from the current task based on the user's knowledge. The user is shown each question individually. The student is asked to click enter after typing the answer in. The ontology checks for the appropriate action and takes corrective action, if necessary, after receiving user input. The user-provided answer is evaluated for accuracy.

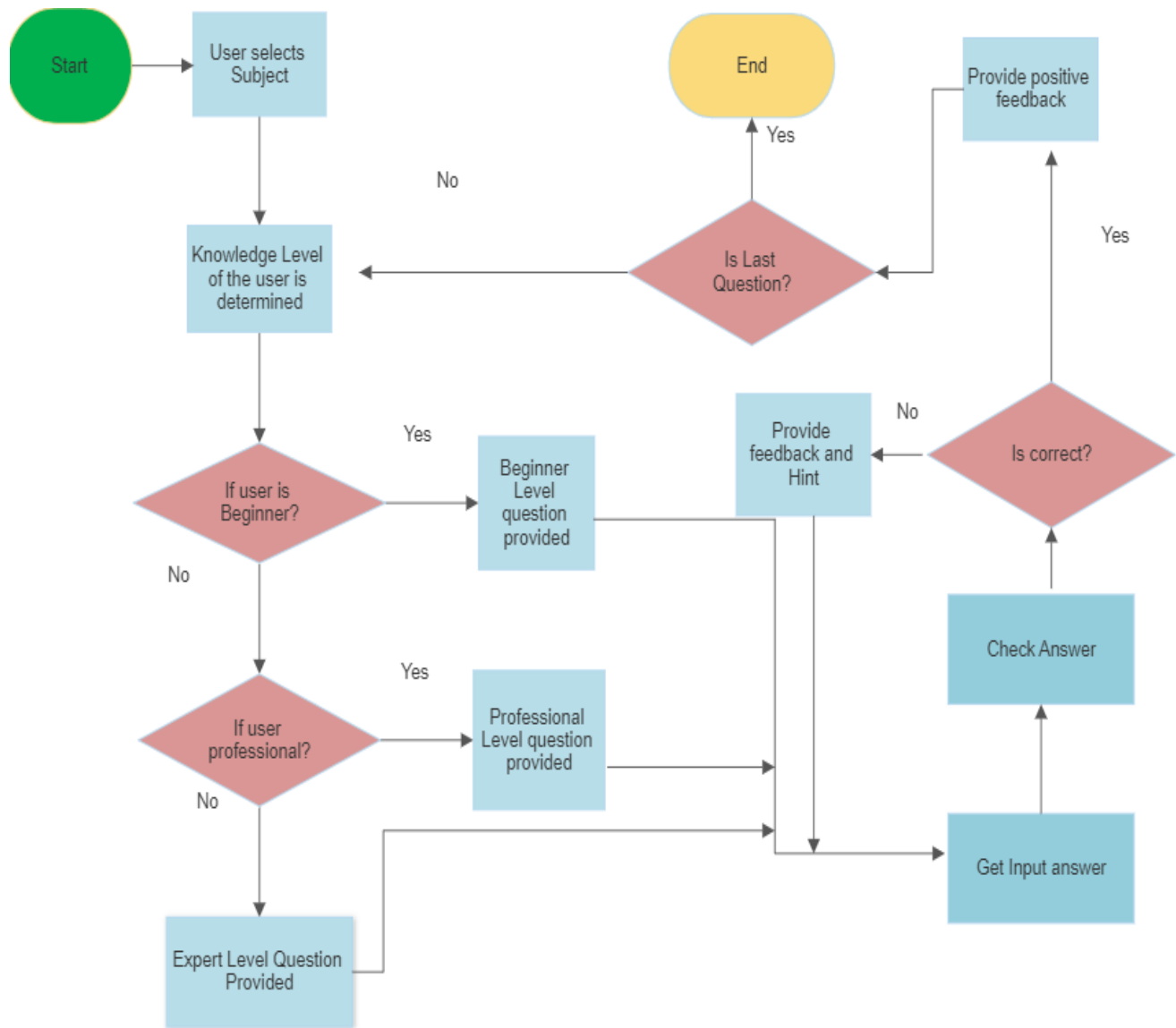


Figure 10 Dialog Actions

If the response is correct, constructive feedback is given to the student. If the student offered an incorrect response, a corresponding negative feedback or statement of motivation along with a hint is provided. Figure 10 illustrates the entire interaction along with the conditional actions.

4.4.4 Q-Learning for Self-Learning of POMDP Policy

Reinforcement Learning (RL) is a form of Machine Learning (ML) where the system learns based on performing actions in an unknown space, observing a reward, and then using the observed reward to inform future action choices [40]. Also, the new mode is passed so that it will be used for future rewards. The reward and mode transfer that occurred affects the choice of the action taken while choosing that mode.

For example, for the question provided by the system about “Why does Canada have so many different climates?”, the user may provide the answer “size”. That text passes to the belief state calculator. The calculator will compare each word to the tags of the intents, and in this case, only small-size and large-size intents are relevant. They both contain the tag “size” within the user input, which significantly increases their belief. Large size contains the tag “large landscapes” and small tag “small landscapes” as these words are like the word size but not exact. As the large landscape is closer than the small landscape, the former receives 0.870 as its value of belief state and the latter receives 0.800. These values represent the belief state that the system adds to the BSH. The simplified belief state is $(0.870 + 0.800)/2 = 0.835$. With an assumption that there are three data points in the BSH, the conversation continues as illustrated in Table 8 until the user provides a correct answer to the given question.

User Input	Large Landscape	Small Landscape	NCP	Knowledge level	System Reply
Size	0.870	0.700	0	Expert	Please try again!

Because of its size	0.880	0.240	1	Expert	Please try again!
Large size	0.897	0	2	Expert	Correct Answer

Table 8 Example Conversation History

The system employs the updated Q-Learning equation below to take the next action.

$$Q(x, a) = (1 - \alpha) Q(x, a) + \alpha (r + \gamma \max_b (Q'(y, b)))$$

In the above equation, Q' is the new mode. This technique allows the PODMP policy to self-adjust.

For example, the system consists of ten states and three actions, assuming that the system is in state 4 and the best action is 3. Table 9 contains related q values. This example will use a reward discount of 0.5 and a learning rate of 0.8.

State	Action 1	Action 2	Action 3
1 = 0.0-0.1	2	7	5
2 = 0.1-0.2	5	4	9
3 = 0.2-0.3	9	2	3
4 = 0.3-0.4	2	2	7
5 = 0.4-0.5	5	2	8
6 = 0.5-0.6	1	2	4
7 = 0.6-0.7	4	8	1
8 = 0.7-0.8	4	2	9

9 = 0.8-0.9	3	8	2
10 = 0.9-1.0	2	4	6

Table 9 Self-learning of Policy Example

In the example given in Table 8, the user has provided "Large landscape" as a perfectly valid response. This observation will move the belief state to 0.880 for "Large Landscape" and 0 for "Small Landscape", creating a reward of 0.242. The new simplified belief state is 0.880, as it is the only belief which is greater than zero. Now the system will calculate the NCPs by passing the BSH, including this input, into the DWT to get the transformed wave. Then each zero-crossing point is counted to get the total NCPs. Figure 11 shows the waves made by the BSH, while Figure 12 is the resultant DWT wave.

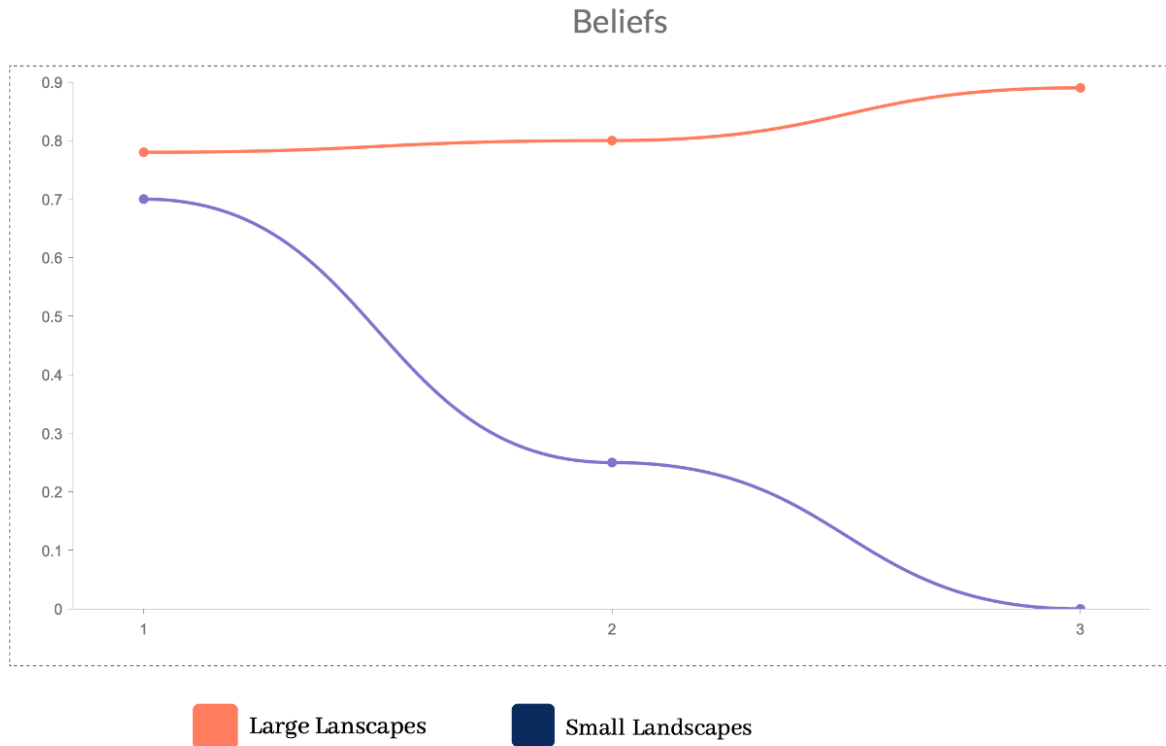


Figure 11 Belief State History Example

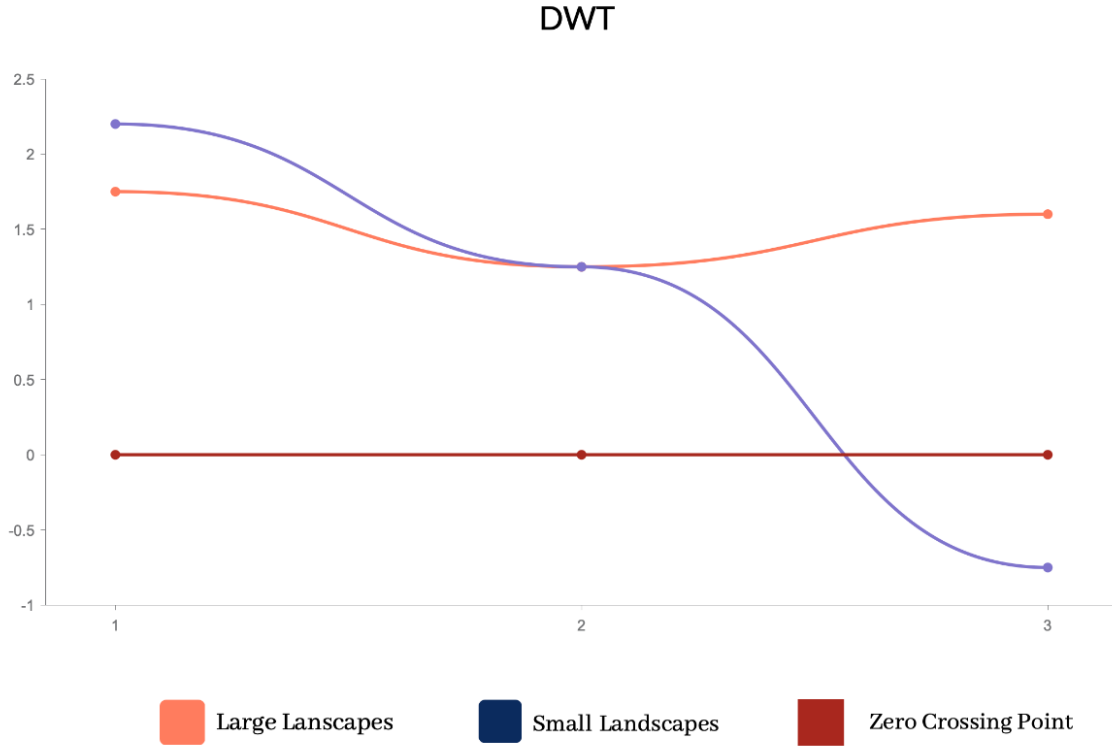


Figure 12 Transformed DWT Example

The total number of zero-crossing points is one, which corresponds to the expert user. From Table 9, the best action for state 8 is again action 3. Therefore, the system will say, “Correct Answer Keep Going”. While this response does not make much sense based on the belief states, as q-learning keeps going, this would ideally improve.

Now that the system has a reward and its new mode, it will update the state 4 action 3. Based on this data, the system updates the q-value for state 4 action 3 using equation.

$$Q(4,3) = (1 - 0.8)Q(4,3) + 0.8(3 + 0.5 \max_b(Q'(8,b)))$$

The best action for state 8 is action 3, which has a q-value of 9.

$$Q(4,3) = (0.2)(7) + 0.8(3 + 0.5(9))$$

$$Q(4,3) = 7.4$$

Table 10 shows how state 4 action 3 is updated:

State	Action 1	Action 2	Action 3
4 = 0.3-0.4	2	2	7.4

Table 10 Self-learning of Policy Updated q table.

Until the system reaches an end state and closes, this procedure is repeated. In q-learning, the final state is decided externally rather than stated.

4.4.5 Q-Learning for Self-adjusted User Clustering

The system uses Q-learning to adjust the NCP ranges and cluster users into expert, professional and beginner. Q-learning will be performed in the same way as discussed in Subsection 4.4.4.

For example, the system consists of 3 states (clusters) and 10 actions (NCP) and Table 11 contains related q values. This example also uses a reward discount of 0.5 and a learning rate of 0.8.

State	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
(Cluster)	NCP	NCP	NCP	NCP	NCP	NCP	NCP	NCP	NCP	NCP
	1	2	3	4	5	6	7	8	9	10

1 Expert	4	9	1	3	6	4	3	4	6	7
2 Professional	2	3	4	7	9	8	7	5	2	1
3 Beginner	1	4	3	2	3	6	5	7	8	9

Table 11 q table for Self-adjusted User Clustering

Suppose the system is in state one. With the help of the q values the system gets rewarded for classifying the type of user with the NCP. Looking at Table 11, the best action is action 2 as it has a q-value of 9, which is greater than other actions and classifies the user as “Expert”. Therefore, the system proceeds with action 2. After receiving a reward of 3, the system transits to state 2. Based on this data, the system updates the q-value for state 1 action 2 using the following equation.

$$Q(1,2) = (1 - 0.8)Q(1,2) + 0.8(3 + 0.5 \max_b(Q'(2,b)))$$

The best action for state 3 is action 1, which has a q-value of 6.

$$Q(1,2) = (0.2)(9) + 0.8(3 + 0.5 (9))$$

$$Q(1,2) = 7.8$$

Table 12 illustrates the updated q table after several iterations on the experimental dataset. Based on the q-table the right range of NCP for different types of users is obtained, i.e., [0, 3] for experts, [4, 7] for professionals, and [8, 10] for beginners.

State (Cluster)	A1 NCP 1	A2 NCP 2	A3 NCP 3	A4 NCP 4	A5 NCP 5	A6 NCP 6	A7 NCP 7	A8 NCP 8	A9 NCP 9	A10 NCP 10
1 Expert	8	8.5	8	3	6	4	3	4	2	2
2 Professional	2	3	4	7	9	8	9	5	2	1
3 Beginner	1	4	3	2	3	6	5	9	9	9

Table 12 Updated q table for Self-adjusted User Clustering

4.4.6 ECA Simulation

The simulation of ECA relies on Sentiment Analysis, fuzzy logic, forced phonetic alignment, emotion, and facial expression. Sentiment analysis is based on fuzzy logic. This emotion defines the facial emotion which is seen on the ECA. The sentiment analyzer uses xforced phonetic alignment after determining the action and emotion. It returns a schedule that tells the ECA simulator which phonetic syllable to simulate at each point in time. The audio generated by the text-to-speech engine is merged with emotion and facial expression. The ECA receives this audio as its last input and produces the simulation by playing the audio file with synchronized lip movements and facial expressions.

4.4.7 Ontology Result

A single transaction is kept in the ontology database with all the data from the interaction, including the user input, the details of the question, and the accuracy of the answer. Upon the completion of all these steps, the system chooses and displays

the next question depending on the learner's updated knowledge level, and the process described above is repeated. The student will have a better understanding of a particular subject once they have answered all the questions for the current task, at which point they can go on to the next subject.

4.5 Running Examples

This section goes over three distinct cases of personalized tutoring with and without self-adjusted NCP for user clustering in order to further explain the suggested methodology.

EXAMPLE 1

This example shows how a learner interacts with the system in a linear learning environment without personalization of questions in accordance with the user's knowledge level.

Current task details:

Subject: Math

Number of questions: 20

Level of the question: Expert

Knowledge level of user: Beginner

Current State of student:

Subject: Math

Question: Is $2 > 5$?

Correct Answer: No

User Attempt: 0

Without personalization, the learner will continue to the last question of the task without understanding or reviewing the concept in which he or she is making mistakes or getting positive feedback for ideas that he or she has learned effectively, as shown in the table (with feedback). Table 13 illustrates the “Example 1”.

Question No	Is correct?	Action
1	True	Correct Answer
2	False	Next question
3	False	Next question
4	False	Next question

Table 13 Sample Interaction Example 1

Hints and feedback could have been given during the interaction, but without any adjustment to the level of difficulty because the system is unaware of the user's knowledge level.

EXAMPLE 2:

This example demonstrates how a learner interacts with a personalized tutoring system with self-adjusted(Q-learning) NCP ranges to generate professional-level questions if the user type is classified as professional.

Current task details:

Subject: History

Number of questions: 20

Level of the questions: Professional

Knowledge level of user: Professional

In Example 2, it is assumed that the user provides wrong answers to Questions 1-2 in a row. As the user starts at the professional level, the system asks two professional questions and provides hints at the same level as well. However, two consecutive mistakes bring the user's level from "Professional" to "Beginner". For Questions 3-8 at the beginner level, the user manages to answer them all correctly. These six consecutive correct answers, without the need of any hints, bring the user's level of knowledge back from "Beginner" to "Professional".

For Questions 9-12, the user needs to answer at least three consecutive questions correctly to remain in the professional level but manages to answer two correctly, which results in a drop of knowledge level again to "Beginner". As this is a professional user, he/she is expected to answer Question 13-20 correctly, and six correct answers in a row help to restore his/her knowledge level to "Professional".

Table 14 demonstrates a sample interaction.

Question no	Is correct?	Knowledge Level	Action
1-2	False	Beginner	Incorrect. Please try again.
3-8	True	Professional	Correct answer!! Go to next question
9-12	False	Beginner	Incorrect. Please try again.
13-20	True	Professional	Correct Answer!!

Table 14 Sample Interaction: Example 2

EXAMPLE 3:

This example shows how self-adjustment of the NCP range may be used to assess the user's understanding of each question throughout the entire subject in order to choose the subsequent question for the user in accordance with the learner's knowledge level considering the learner to be an expert.

Let a user start at the expert level when working on a set of 20 questions on the subject of geography and allow the user to give correct answers to Questions 1-8 in a row. As a result, the user's knowledge level remains unchanged at the end of Question 8. For Questions 9-12, the user needs to answer at least three consecutive questions correctly, but the user's ability to answer only two correctly brings his/her knowledge level down to "Professional". For the remaining questions, six correct answers to Questions 13-20 in a row bring the knowledge level back to "Expert". During the interaction, hints are provided according to user's level of knowledge.

In Example 2 and Example 3, users are chosen from the dataset as experts, professionals, and beginners if they can respectively provide more than 80%, between 51%-79%, or less than 50% correct answers to 20 questions on a subject. During interactions, their knowledge levels are determined dynamically at runtime according to the self-adjusted user clustering. Table 15 demonstrates a sample interaction.

Question no	Is correct?	Knowledge Level	Action
1-8	True	expert	Correct answer!!

			Go to next question
9-12	False	Professional	After multiple attempts! Go to next question
13-20	True	Expert	Next question

Table 15 sample interaction for Example 3

Chapter 5

Implementation and Experiments

Details of the experiments and implementation are provided in this chapter. The core concept of the thesis method is self-adjustment of POMDP policies and user clustering. The enhancement introduced by this thesis research helps to improve the performance of the personalized tutoring system.

5.1 Software and Tools

The following table contains a comprehensive list of the libraries, tools, software, languages, etc.

Function	Software
Programming Language	Python
Avatar Simulation	AirSim
Ontology Language	OWL
Ontology Editor	Protégé
Ontology Manipulation from Python	owlready2
Phonetic Alignment	PocketSphinx
Text-to-speech	Google Text-to-speech
Code Editor	Anaconda Spyder
Dataset format	RDF/XML
Sentiment Analyzer	nltk.sentiment.vader

Table 16 Software and Tools

5.2 Dataset

The dataset utilized for domain knowledge and references is from [21]. In order to improve the data quality, the data is first cleaned. The experiment's domain is elementary school subjects, and the dataset has 20 questions in each of the four subjects of Articles, Math, History, Geography, and Science.

The dataset is designed in a question-and-answer format where students can provide concise answers to the questions. All questions in the dataset are identified by numbers, categorized into subjects, and provided with correct answers. In addition, each question has a tag to indicate the associated level of knowledge. Upon preprocessing and cleaning, the dataset yields the following data that is directly applicable to the algorithm:

Question No	Subject	Question	Answer	Knowledge Level
-------------	---------	----------	--------	-----------------

Table 17 Components of the Datasets

5.3 POMDP Policy Selection

The experiments use three policies, one for each of the three knowledge levels for beginner, professional, and expert. The experiment is carried out for a single RL policy because all three machine learning policies are equivalent. There is no need to use all three as q-learning tends towards the optimal policy [43], and the reward is based on the likelihood of choosing the best policy. The q-learning policies are hand-crafted at first, but rather than using pre-selected boundaries, they choose the action by q-learning. The value of the belief state ranges between zero and one, and the belief state can be equally divided into 10 sub-spaces, each of which correlates with a state. This leads to a ten-state, three-action, q-value matrix as there are three actions. After each conversation, all values are stored and initialized to 0. At the beginning, all three policies are the same, but as training proceeds, they will diverge.

If the policy being used changes, the system will consider the new policy in the future to determine the reward value.

The increase in the average non-zero belief divided by the number of non-zero intents determine the reward given by the change in belief state. The following reward formula is motivated by the notion that the system aims to have a single intent with a high belief. The q-values for q-learning are trained by using this reward.

$$\text{reward} = \frac{\text{average}(B) - \text{average}(B')}{|B|}$$

Experiment results from [11] had shown that employing the self-adjusted policies leads to shorter interactions and, as a result, provides a better user experience. A similar outcome is anticipated for a personalized tutoring system as the enhancement with ML leads to shortened average dialog lengths.

5.4 Ontology Structure

The construction of an ontology using Web Ontology Language (OWL) is explained in depth in this section using the demo dataset. Each class is referred to as a subclass of "owl:Thing". Ontology in OWL can be developed in Protégé. The "owl:thing" ontology features three subclasses, including ECATutor, ECASStudent, and ECASubject. Moreover, ECASubject includes subclasses for different subjects, including Articles, Math, History, Geography, and Science.

5.4.1 Visualization in Protege:

The construction of ontology uses Protégé's ontology editor and framework, which are both open-source and free. Figure 14 shows the Protégé implementation of the class hierarchy and the individual relationships between them.

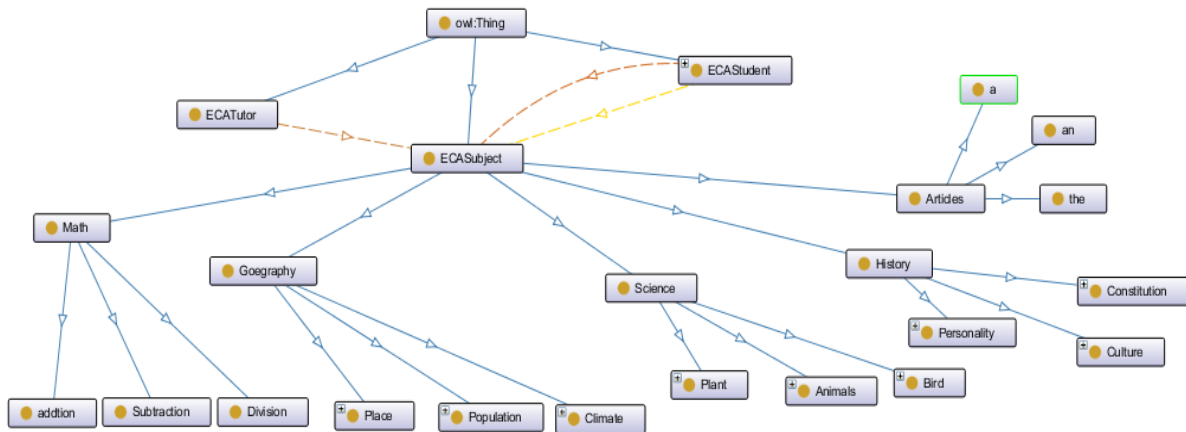


Figure 13: OWL Class Structure

The classes' corresponding object properties, as well as the domain and range of each object property, are denoted by the arrows between them. Table 18 demonstrates a sample ontology instance for the subject geography. It includes a question, a possible correct answer, level of knowledge for “Beginner”, hint and feedback.

Subject: Geography	
Question	Why does Canada have so many different climates?
Possible Correct answer	Because of its size, Canada has many different climates
Knowledge Level	Beginner
Hint	Large landscape
Feedback	Well done!!

Table 18 Instance 1: Geography

The classes, data properties and object properties are shown below.

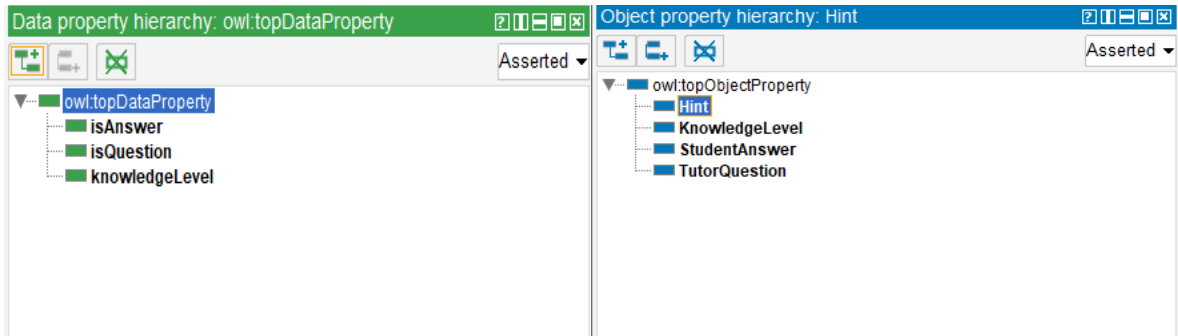


Figure 14 OWL Properties

5.5 User clustering

During each interaction loop, the knowledge level of each student is determined at runtime in order to give the user a personalized learning experience. This allows the system to give the user the best questions for their level of understanding in each subject. Instead of using hand-crafted NCP values to assess a learner's level of understanding, a self-adjustment methodology based on q-learning is implemented. A more effective strategy would be to use grid searches to identify the best knowledge level boundaries. While grid searches need a lot of computation time, this approach would get the best result.

Grid search ran with 2000 training samples at each combination. As 2000 conversations is a substantial number, a more extensive test is not required. The system is training four times as many q-values, hence the increase in training samples is justified. The learning rate was 0.8 during training and 0.5 during testing, with all q-learning reward discounts set to 0.5. In order to determine the q-learning learning rate (α) and reward discount (γ) experiments were conducted using different values. For the learning rate, it was found that lowering the learning rate amid testing had better results. For the discount factor rate, similar adjustments were made, including using different learning rates for each policy. In all cases, the finest

result came from using the reward discount of 0.5. The ultimate NCP ranges were between 1 and 3 for an expert, between 4 and 7 for a professional, and between 8 and 10 for a beginner.

Chapter 6

Experimental Results

Each interaction between the tutoring system's question and the learner's answer ends in one of the following three outcomes:

1. The tutor offers encouraging feedback.
2. The tutor provides the student with a hint so they can respond correctly.
3. When the student provides a correct response, the tutor goes on to the following question based on the knowledge level.

The experimental application of the tutoring model outlined in this thesis may be used to gather a wide range of information regarding students, tutoring techniques, and subject areas. Additional possible use includes data analysis for trends, refinement of practical teaching methods, perception of the psychology of the students based on the responses, and changes in performance before and after feedback.

Based on the dataset received from [21] and the use of this strategy, the following major results were produced.

6.1 User Clustering Results

Table 19 is the result of a comparison of how the users are classified. The classification of 20 users who attempted questions on geography is shown in the second column. The system uses q-learning to classify the results, considering the user's knowledge and the correctness of the answer. The system assumes that users, who receive positive feedback, gain more confident about a specific subject. Alternatively, users who needs hints to produce correct answers gain knowledge

about the subject. Information on the learners' classification using handcrafted values (DWT) is provided in the third column.

Knowledge level	User(q-learning)	Users (DWT)	Accuracy
Expert	10	9	85%
Professional	6	7	75%
Beginner	4	5	90%

Table 19 User Clustering

While the dataset includes information about the percentage of correct answers, the experiment uses this information to classify users as "Expert", "Professional", and "Beginner" if they received a grade of at least 80%, between 51% to 79%, and less than 50% respectively.

The average accuracy when the classification is performed by q-learning is 86.9%. The discrepancy in the number is mostly because the system uses q-learning to choose the best question to ask each user based on their level of knowledge. This enables each user to respond to their own question without the need for hints.

6.2 Personalized Questions and Hints/Feedback

Based upon the dataset in [21], students are more motivated to keep learning a subject when they get personalized questions based on their level of understanding. The graph in Figure 15 illustrates how personalized questions affect learner performance. Data analysis reveals that the learner gradually raises his/her knowledge level after gaining a deeper understanding of the topic.

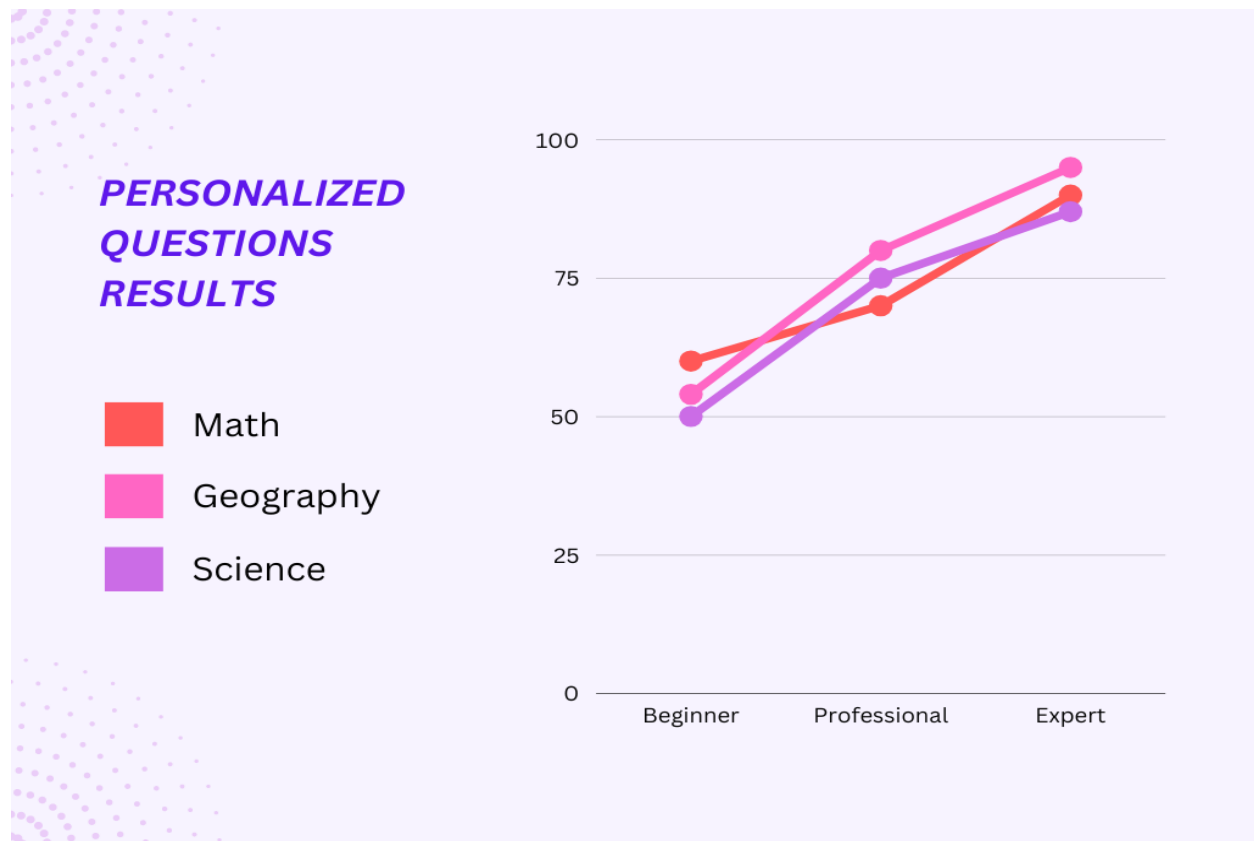


Figure 15 Personalized Questions Results

As it was discussed in previous sections, personalized tutoring changes the question based on the user's knowledge level. Once the user answers the current question, the q-learning algorithm determines the knowledge level responsible for choosing the next question. The students who attempted questions from the geography subject demonstrated the results shown in Table 20.

The knowledge level as assessed by the q-learning classification is shown in the first column. The second column represents how many beginning-level hints were given overall during the 20 students' interactions. The total number of hints for the intermediate and expert levels are shown in columns three and four, respectively. Column five lists the average number for the 20 questions where the learner received

positive feedback. The next question level selected for the students of each class is listed in the last column and is based on the user's updated knowledge.

Knowledge	Total Users	No. of Hints (Beginner)	No. of Hints (Prof.)	No. of Hints (Expert)	Average Feedback
Expert	10	0	0	1	12
Professional	6	5	9	0	9
Beginner	4	4	4	0	6

Table 20 Summary of Personalized Questions and Hints

Only one of the ten students who were classified as "Expert" in Table 19 needed a clue for a single question. Hence, just one expert-level learner needed a hint out of 20 questions for each of the 10 learners (200 questions). This brings the frequency of using a personalized hint to 1/200. Among the six learners classified as "Professional", most of them received hints once or twice. One student needed seven consecutive hints in the first question, following which he was able to correctly answer the rest of the questions. With 18 good feedbacks, this student successfully answered all the other questions, except for the first two questions. The student was able to pass as a "Professional" learner at the end, even though he had one "Professional" and five "Beginner" level questions. This is because he successfully answered the remaining questions. Six students were given 20 questions each, which is 120 questions in total. Personalized hints were utilized 14 times. The ratio of hint transactions to total transactions for the remaining four students shifts to 12/120, except for the mentioned student. Each of the five students who were classified as "Beginner" by the q-learning classification, required at least one "Beginner" level hint. In addition, the learners showed a pattern where most questions required at least

two-thirds of their efforts to be answered correctly. Even though there were few hints, this conduct led to the total class being classified as "Beginner". As a result, the hints were provided eight times for a total of 100 problems. For each student, the ratio is 0.8 for 20 questions.

A significant observation is that "Professional" level learners needed more help and feedback than "Beginner" level learners to correctly answer problems. This may be explained by the fact that professional-level students could answer most of the questions correctly on their first try, thus frequently encouraged with positive feedback. Most of the 20 questions were answered incorrectly by students classified as "Beginner". They made errors more frequently, and rarely received positive feedback.

6.3 Success Rates

The success rate is a measure of the percentage of times the system correctly identified the user's knowledge level, and it can be used as the main metric to evaluate the system's performance. To determine success, it is important to know the number of right questions provided by the system according to the knowledge level of the user. The system fails if it couldn't provide questions according to the exact knowledge level. For instance, if a user's knowledge level is "Beginner" and the system provides questions at the "Professional" level, it is a failed attempt for the system to ask a question at the right level of knowledge. Hence, an increase in success rates indicates a system improvement.

$$\text{Success Rate} = \frac{\text{Number of successful identifications of user's knowledge level}}{\text{Total number of attempts}}$$

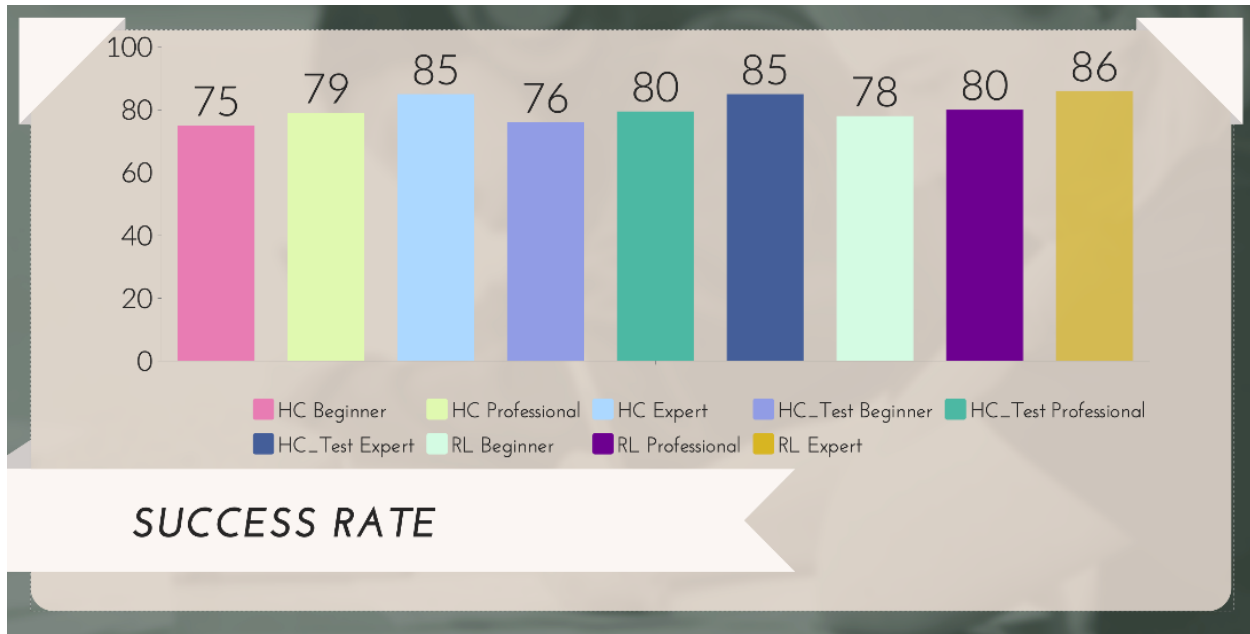


Figure 16 Success Rate Result

Figure 16 provides a visual comparison of the success rates for the three types of users with Handcrafted, Handcrafted_Test(Expert = NCP(1,2), Professional = (3,8), Beginner = (9,10)) and q-learning method for user clustering. The figure shows the success rate is the lowest for "Beginner" classified with hand-craft user clustering. This rate is increased by 3% to 78% after using self-adjusted user clustering. Similar improvement in success rate also appears for "Professional" and "Expert", and the difference between the two user types is 6% with or without q-learning. Overall, the highest accuracy rate of 86.23% was achieved by self-adjusted "Expert". The numbers demonstrate the effectiveness and viability of employing q-learning despite good, handcrafted classification. With q-learning, the questions are more personalized, which helps to boost students' understanding of the subjects.

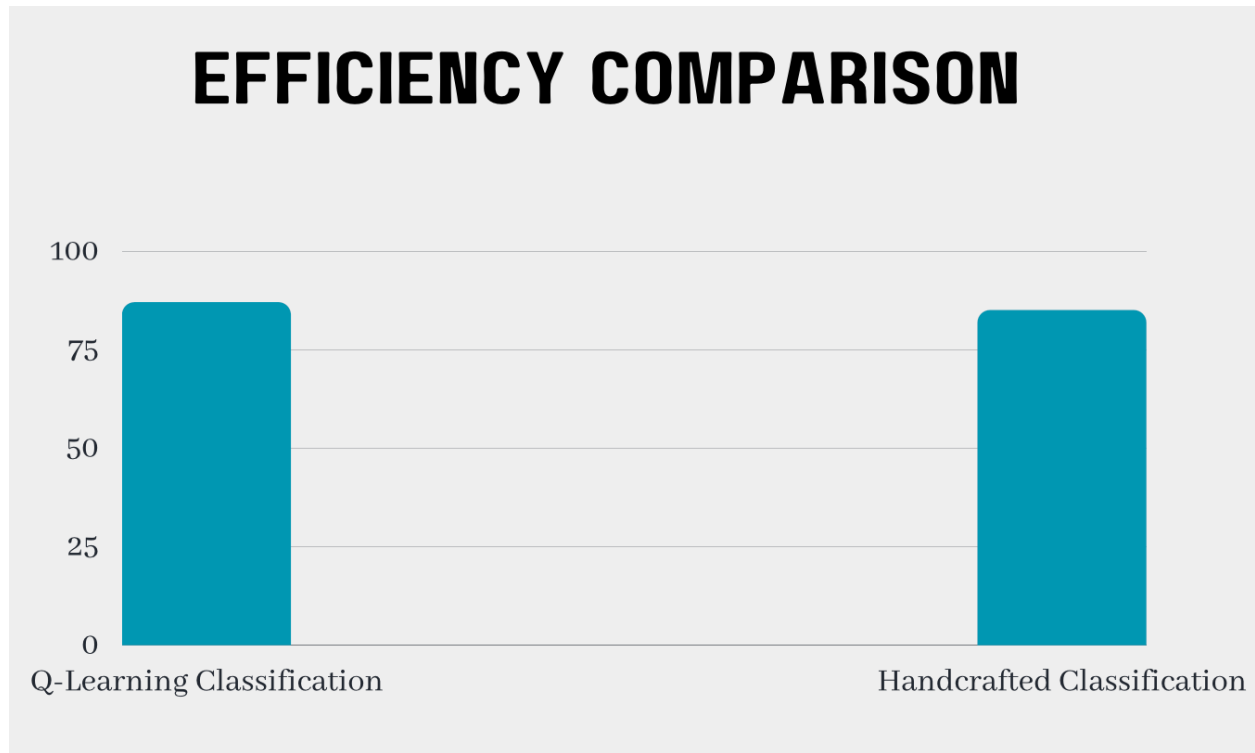


Figure 17 Efficiency Comparison

So, it is determined that Q-learning is more effective with 86.26% when compared to handcrafted classification based on experiments and test scenarios.

RL changed only the NCP ranges for “Expert” and “Professional” types of users, which resulted in improvement in the two ranges. In comparison, there is no change in success rate for "Beginner" type of users as there was no change to the NCP ranges for this type. These findings demonstrate that self-adjusting policy approaches are flawless. The handcrafted values aid to improve RL approaches' overall efficiency and success rate.

6.4 Limitations

There are several limitations on how the entire model can operate, despite the ontology's significant level of adaptability and ability to be merged into any domain. The main limitation is due to the fact that there aren't any publicly accessible datasets

for tutoring system dialogue. The easiest method to train the system would be to put it in a real-world learning environment and record the live tutoring session to add more guidelines and standards for evaluation. In addition, the grid searches on additional hyper-parameters were limited in scope, and they could have missed a result that would have shown an increased improvement. The ECA itself was not tested on additional test subjects to create a Mean opinion score (MOS) on its interactivity. As a result, personal interpretations of concepts like realism or naturalness may differ when presented to a larger audience.

Chapter 7

Conclusion

7.1 Research Summary

This thesis research uses the self-adjustment of POMDP policies and user clustering to enhance the personalized tutoring system. To make the ECA presenting behaviour as human-like as feasible, additional functions can be added to the system as enhancements, such as emotion detection, lip synchronization, and text-to-speech capability customization.

Using a quantitative experiment, this research first illustrates how the use of RL, in the form of Q-Learning, helps to create self-adjusted POMDP policies and to ask the right level of questions and provide the right types of hints with self-adjusted NCP ranges. The experiments conducted revealed a rise in success rates and an enhancement in the system to provide a personalized learning experience. The current implementation totally replaces the original ontology used in the earlier implementation with the more generic educational ontology. This tutoring ontology is effectively designed and implemented in the thesis, allowing the tutoring system to communicate with the users in a more natural way of question-answer interaction. The self-adjustment of the NCP ranges enables the system to more accurately assess the user's level of understanding and personalize questions accordingly. As a result, it helps in building the confidence of the user in learning the subject which in turn improves the success rate.

7.2 Future Research

There are several ways in which this work might be expanded, some of which are described below:

- The tutor may be expanded to include more domains.
- This thesis research extended the quiz style to a question-answer style of interaction, but the ontology may be expanded to include a range of assessments that also include the summary of the answer.
- If more research is necessary, it may be determined by conducting another usability survey to see whether users believe the system appears natural or too robotic.
- To detect the ECA's emotions using RL techniques.

APPENDIX A

Useful Links

Below are the links referred to in the thesis research:

1. [Complete dataset](#)
2. [ECA output videos](#)

REFERENCES

- [1] Niyati Vyas. An Approach of Using Embodied Conversational Agent for Personalized Tutoring. Master's thesis, University of Windsor, 2022.
- [2] H. Ji, I. Han, and Y. Ko, "A systematic review of Conversational AI in language education: Focusing on the collaboration with human teachers," *Journal of Research on Technology in Education*, vol. 55, no. 1, pp. 48–63, 2022.
- [3] Yang, H. and Kyun, S. "The current research trend of artificial intelligence in language learning: A systematic empirical literature review from an activity theory perspective," *Australasian Journal of Educational Technology*, pp. 180–210, 2022.
- [4] Shahriar, Asif & Weber, Peter. "Talk the Talk: Enhancing the Educational Chatbot Experience by Conversational Design", 2022.
- [5] K. Papadopoulos, A. M. Velentza, P. Christodoulou, and N. Fachantidis, "Social Educational Robotics Application: Architecture and interconnectivity," 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA), 2022.
- [6] L. N. Paschoal, T. U. Conte, and S. do Souza, "On the experimental process in evaluations of Brazilian conversational agents in Education," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 17, no. 1, pp. 99–107, 2022.
- [7] Zhihui Wang, Jianrui Chen, Fernando E. Rosas, Tingting Zhu, A hypergraph-based framework for personalized recommendations via user preference and dynamics clustering, *Expert Systems with Applications*, Volume 204, 117552, ISSN 0957-4174, 2022.
- [8] Hwang, Gwo-Jen, Tu, Yun-Fang et Tang, Kai-Yu "AI in Online-Learning Research: Visualizing and Interpreting the Journal Publications from 1997 to 2019". *International Review of Research in Open and Distributed Learning* 23, no. 1: 104–130, 2022.

- [9] Hwang, Wu-Yuin, and Rio Nurtantyana. "X-Education: Education of All Things with AI and Edge Computing—One Case Study for EFL Learning." *Sustainability* 14, no. 19 12533, 2022
- [10] T. Karakose and M. Demirkol, "Exploring the emerging covid-19 research trends and current status in the field of education: A bibliometric analysis and knowledge mapping," *Educational Process: International Journal*, vol. 10, no. 2, 2021.
- [11] Tristan Szucs. Lip Synchronization for ECA Rendering with Self-Adjusted POMDP Policies. Master's thesis, University of Windsor, 2020.
- [12] Hamdi Serin. Virtual reality in education from the perspective of teachers. *Amazonia investiga*, 9(26):291–303, 2020.
- [13] Ryan Lege and Euan Bonner. Virtual reality in education: The promise, progress, and challenge. *JALT CALL Journal*, 16(3), 2020.
- [14] A. K. Das, S. Kalam, N. Sahar, and D. Sinha, "UCFL: User categorization using fuzzy logic towards PUF based two-phase authentication of fog assisted IOT devices," *Computers & Security*, vol. 97, p. 101938, 2020.
- [15] Maiga Chang, Giuseppe D'Aniello, Matteo Gaeta, Francesco Orciuoli, Demetrios Sampson, and Carmine Simonelli. Building ontology-driven tutoring models for intelligent tutoring systems using data mining. *IEEE Access*, 8:48151–48162, 2020.
- [16] Ruturaj Raval. An improved approach of intention discovery with machine learning for pomdp-based dialogue management. Master's thesis, 09 2020.
- [17] Chiatoh, Blasius Agha-ah, and Jude Chia. "The Covid-19 pandemic and the challenge of teaching English online in higher institutions of learning in Cameroon." *Journal of English Language Teaching and Applied Linguistics* 2, no. 5: 35-42, 2020.

- [18] Stancin, K., Posic, P. & Jaksic, D. Ontologies in education – state of the art. *Educ Inf Technol* 25, 5301–5320 , 2020.
- [19] F. den Hengst, M. Hoogendoorn, F. van Harmelen and J. Bosman, "Reinforcement Learning for Personalized Dialogue Management," *arXiv*, 2019.
- [20] F. Pusse and M. Klusch, "Hybrid Online POMDP Planning and Deep Reinforcement Learning for Safer Self-Driving Cars," 2019 IEEE Intelligent Vehicles Symposium (IV), 2019.
- [21] Reddy, S., Chen, D. and Manning, C.D., Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7, pp.249-266, 2019.
- [22] Diego Dermeval, Ranilson Paiva, Ig Ibert Bittencourt, Julita Vassileva, and Daniel Borges. Authoring tools for designing intelligent tutoring systems: a systematic review of the literature. *International Journal of Artificial Intelligence in Education*, 28(3): 336–384, 2018.
- [23] Microsoft, "Home - Airsim," Microsoft, [Online]. Available: <https://microsoft.github.io/AirSim/>, 2018.
- [24] X. Yuan, M. Kaler and V. Mulpuri, "Personalized visualization based upon wavelet transform for interactive software customization," *Machine Learning and Data Mining in Pattern Recognition*, pp. 361-375, 2017.
- [25] Bih Ni Lee, Sopia Abdullah, and Su Na Kiu. Supportive learning: Linear learning and collaborative learning. *Turkish Online Journal of Educational Technology-TOJET*, 15(2):97–105, 2016.
- [26] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep Q-learning with model-based acceleration," *arXiv.org*, [Online]. Available: <https://arxiv.org/abs/1603.00748> , 2016.

- [27] Vijaya Krishna Mulpuri. Trend Analysis of Belief-State History with Discrete Wavelet Transform for Improved Intention Discovery. PhD thesis, University of Windsor (Canada), 2016.
- [28] P. Gabriel, J. Nielsen and P. RIGO, "POMDP-Based Risk Maintenance Planning for Offshore Wind Substructures," 2016.
- [29] R. Luckin, L. B. Forcier, G. Mark , and H. Wayne , Intelligence Unleashed: An argument for AI in Education. Pearson Education, 2016.
- [30] Shubhrendu Tripathi, A Run-Time Approach of Combining Ontologies to Enhance Interactive Requirements Elicitation for Software Customization, University of Windsor, 2016.
- [31] X. Yuan and X. Zhang, "An ontology-based requirement modeling for interactive software customization," in 2015 IEEE International Model-Driven Requirements Engineering Workshop (MoDRE), Ottawa, ON, Canada, 2015.
- [32] X. Yuan and R. Vijayarangan, "Emotion Animation of Embodied Conversational Agents with Contextual Control Model," in 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, 2013.
- [33] Milica Gašić and Steve Young. Gaussian processes for pomdp-based dialogue manager optimization. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(1):28–40, 2013.
- [34] Patrick, Susan, Kathryn Kennedy, and Allison Powell. "Mean What You Say: Defining and Integrating Personalized, Blended and Competency Education." International Association for K-12 Online Learning , 2013.
- [35] R. W. Lawler and N. Rushby, "An interview with Robert Lawler," British Journal of Educational Technology, vol. 44, no. 1, pp. 20–30, 2012.
- [36] Guoying Liu, A task ontology model for domain independent dialogue management, University of Windsor, 2012.

- [37] I. Magnisalis, S. Demetriadis, and A. Karakostas, "Adaptive and Intelligent Systems for Collaborative Learning Support: A Review of the field," *IEEE Transactions on Learning Technologies*, vol. 4, no. 1, pp. 5–20, 2011.
- [38] Songjie Gong. A collaborative filtering recommendation algorithm based on user clustering and item clustering. *J. Softw.*, 5(7):745–752, 2010.
- [39] "POMDPs(=MDPs+HMMs)," [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.187.7395&rep=rep1&type=pdf>, 2010.
- [40] X. Yuan and L. Bian, "A modified approach of POMDP-based dialogue management," 2010 IEEE International Conference on Robotics and Biomimetics, Tianjin, China, 2010.
- [41] Sabiha Sathulla. Four mode-based dialogue management with modified pomdp model. Master's thesis, University of Windsor, 2010.
- [42] B. P. Woolf, "A roadmap for education technology - CRA," [Online]. Available: <https://cra.org/crc/wp-content/uploads/sites/2/2015/08/GROE-Roadmap-for-Education-Technology-Final-Report.pdf>, 2010.
- [43] Danuta Zakrzewska. Cluster analysis in personalized e-learning systems. In *Intelligent systems for knowledge management*, pp. 229–250. Springer, 2009.
- [44] G. Hollinger, "Partially Observable Markov Decision Processes (POMDPs)," [Online]. Available: https://www.cs.cmu.edu/~ggordon/780fall07/lectures/POMDP_lecture.pdf, 2007.
- [45] Jason D Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.
- [46] Mahamed GH Omran, Andries P Engelbrecht, and Ayed Salman. An overview of clustering methods. *Intelligent Data Analysis*, 11(6):583–605, 2007.
- [47] Kurt VanLehn. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265, 2006.

- [48] M. Rehm and E. André, "From annotated multimodal corpora to simulated human-like behaviors," *Modeling Communication with Robots and Virtual Humans*, pp. 1–17, 2006.
- [49] Jokinen, Kristiina. "Challenges for adaptive conversational agents." In *Proceedings of the Baltic Conferences on Human Language Technologies*, pp. 51-60. 2005.
- [50] Hartmann, B., Mancini, M. and Pelachaud, C., January. Towards affective agent action: Modelling expressive ECA gestures. In *International conference on Intelligent User Interfaces-Workshop on Affective Interaction*, San Diego, CA, 2005.
- [51] June Lu, Chun-Sheng Yu, and Chang Liu. Learning style, learning patterns, and learning performance in a webct-based mis course. *Information & Management*, 40(6): 497–507, 2003.
- [52] Natalya F Noy, Deborah L McGuinness, et al. *Ontology development 101: A guide to creating your first ontology*, 2001.
- [53] Fry, Kate. "E-learning markets and providers: some issues and prospects." *Education+ Training* 43, no. 4/5,233-239, 2001.
- [54] Justine Cassell. Embodied conversational agents: representation and intelligence in user interfaces. *AI magazine*, pp. 22(4):67–67, 2001.
- [55] C. J. Watkins and P. Dayan, "Technical Note Q-Learning," *Machine Learning*, vol. 8, no. 3, p. 279–29, 1992.

VITA AUCTORIS

NAME: Ashwitha Vichuly Jawahar

PLACE OF BIRTH: Chennai, India

YEAR OF BIRTH: 1996

EDUCATION: Anna University, B.Tech, Chennai, India, 2018
University of Windsor, M.Sc., Windsor, ON,
2023