



**KENYATTA UNIVERSITY**

**DEPARTMENT OF COMPUTING & INFORMATION TECHNOLOGY**

**AN INTELLIGENT BASED SYSTEM FOR SUPPORTING PERSONALISED**

**E-LEARNING**

**KIVUVA FAITH NGAMI**

**J57/37484/2017**

**SUPERVISOR: DR.ELIZAPHAN MAINA**

**THIS PROJECT REPORT IS SUBMITTED FOR THE PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTERS OF  
SCIENCE IN COMPUTER SCIENCE IN THE SCHOOL OF ENGINEERING AND  
TECHNOLOGY OF KENYATTA UNIVERSITY**

**FEBRUARY 2023**

## **Declarations**

I declare that this project is my original work and has not been presented in any other university/institution for consideration of any certification. Referenced sources that have been properly acknowledged have been included in this research project. Where text, images, photographs, figures, or tables are taken from other sources, including the internet, they are properly acknowledged and the sources are cited in accordance with anti-plagiarism regulations and the current APA style.

**Signature** \_\_\_\_\_ **Date:** \_\_\_\_\_

Faith Ngami Kivuva

J57/37484/2017

Computing and Information Technology

This project has been submitted for appraisal with my approval as University Supervisor.

**Signature** \_\_\_\_\_ **Date** \_\_\_\_\_

Dr. Elizaphan Maina

Computing and Information Technology

Kenyatta University

## **Dedication**

I dedicate this project to my loving husband Justus and my daughters Hazel and Olive. The project would not have been possible without their support, encouragement, and inspiration.

Thank you and may God bless you.

## **Acknowledgment**

This project's origin, development, and successful conclusion are all due to several people, without the help of whom the intellectual journey would not have been possible. I give thanks to the Almighty God for providing me with the chance, good health, and resources I need to carry out the study. I also want to express my gratitude to my supervisor, Dr. Elizaphan Maina, for his constant support, wise counsel, and willingness to share his extensive expertise and experience with this project.

I want to express my gratitude to my professors as well for the abundance of knowledge they provided me with regarding this study. Finally, but not least, I want to express my gratitude to my classmates for their contributions to the growth of this research.

## **Abstract**

Most traditional e-learning systems fail to provide the intelligence to guide a learner according to their learning style. However, intelligent agents can be created to perform the role of guide to a student depending on a predetermined learning style. In view of this, the study discusses how to design, develop and implement intelligent agents for supporting personalized e-learning based on a predetermined learning style. The main objective of this study was to design and implement an intelligent e-learning system based on intelligent agents for supporting personalized e-learning. The system, which is based on intelligent agents, provides some intelligence and supports dynamic learning. Each learner has different levels of achievement depending on their learning styles and gets personalized feedback/recommendations. Three intelligent agents were developed; a learner agent, a tutor agent, and an information agent. The learner agent, which has an AI engine, uses deep neural networks to provide a recommendation to the learners based on their learning styles. The tutor agent accesses what the learner has accessed and passes this information to the learner agent which then recommends the appropriate materials. The information agent presents the recommendations/feedback of the learners through the Moodle user interface. The learning styles of the students are determined by filling out a Visual, Aural, Read/Write, and Kinesthetic (VARK) questionnaire. The three agents were developed using the Prometheus methodology. They were also tested and integrated into Moodle Learning Management System (LMS). This integration allows learners who are using LMS such as Moodle to learn based on their learning style. The results indicate that it is possible to train a learner agent using deep neural networks and provide personalized learning to the learner based on the learning style. Future studies need to focus on using data collected in a learning management system to identify learner styles instead of using the VARK questionnaire. Additionally, it is necessary to use other learning styles models, such as the Filder-Silverman model, and the Kolb learning style model among others, to identify learning styles and conduct an experimental study to determine their effectiveness in personalized learning with intelligent agents.

## Table of Contents

Declarations .....	i
Dedication .....	ii
Acknowledgment .....	iii
Abstract .....	iv
Table of Contents .....	v
List of Figures .....	xi
List of Abbreviations .....	xii
CHAPTER ONE .....	1
1.0 Introduction.....	1
1.1 Background Information.....	1
1.1.1 Components and Architecture of the E-Learning System.....	2
1.1.2 Personalized Learning Strategies .....	3
1.2 Problem Statement .....	5
1.3 Objectives of the study.....	6
1.3.1 Main objective.....	6
1.3.2 Specific objectives.....	6
1.4 Justification .....	6
1.5 Significance of the study.....	6
1.6 Scope of the study .....	7

CHAPTER TWO .....	8
LITERATURE REVIEW .....	8
2.0 Introduction.....	8
2.1 Intelligent Agents.....	10
2.2 Intelligent Agents Architectures .....	11
2.2.1 Logic-Based architecture.....	12
2.2.2 Reactive Architecture.....	12
2.2.3 Belief-Desire-Intention (BDI) Architecture .....	13
2.2.4 Layered (Hybrid) Architecture.....	14
2.2.5 Cognitive Architecture .....	16
2.3 Methodologies for Developing Intelligent Agents .....	17
2.3.1 The Prometheus Methodology .....	17
2.3.1.1 System Specification Phase.....	17
2.3.1.2 Architectural Design Phase .....	18
2.3.1.3 Detailed Design Phase.....	19
2.3.2 MaSE Methodology .....	20
2.3.2.1 The Analysis Phase.....	21
2.3.2.2 The Design Phase .....	21
2.3.3 Tropos Methodology .....	21
2.3.3.1 Early Requirements Phase.....	22

2.3.3.2 Late Requirements Phase.....	22
2.3.3.3 Architectural Design phase.....	22
2.3.3.4 The Detailed Design phase .....	23
2.3.3.5 The Implementation phase.....	23
2.4 AI Techniques Applied in Intelligent Agent Systems .....	23
2.5 Personalized e-Learning.....	25
2.5.1 Prior Studies on Personalized e-Learning .....	26
2.5.2 Personalized e-Learning and Content .....	29
2.6 Overview of e-Learning Systems Based on Intelligent Agents .....	31
2.6.1 Intelligent Tutoring System.....	31
2.6.2 An Agent-Based Intelligent Tutoring System.....	32
2.6.3 Personalized Intelligent Multi-Agent Learning System.....	34
2.7 Overview of the Different Learning Style models.....	34
2.8 Personalized E-Learning and Intelligent Agents .....	38
2.9 Overview of Moodle Learning Management System.....	39
2.10 Theoretical framework.....	42
CHAPTER THREE .....	45
RESEARCH METHODOLOGY.....	45
3.0 Introduction.....	45
3.1 Research Methodology .....	45

3.2 System specification/Problem definition .....	46
3.3 Architectural Design .....	46
3.4 Detailed Design.....	47
3.5 Implementation .....	48
3.6 Testing.....	49
CHAPTER FOUR.....	50
RESEARCH FINDINGS AND DISCUSSIONS.....	50
4.0 Introduction.....	50
4.1 Determining the Learning Style of the Learner .....	50
4.2 Design and Development of the Intelligent Agents .....	52
4.2.1 The Learner Agent .....	53
4.2.2 Tutor Agent .....	64
4.2.3 Information Agent .....	65
4.3 System Architecture.....	66
4.4 System Integration .....	68
4.5 Results from the Learning management system .....	69
4.6 Discussion of Research Findings .....	73
CHAPTER FIVE .....	75
CONCLUSION AND RECOMMENDATIONS .....	75
5.0 Introduction.....	75

5.1 Summary of Research Findings .....	75
5.2 Conclusion .....	75
5.3 Recommendation of the study .....	76
5.4 Limitation of the Study .....	76
5.5 Further Research .....	77
REFERENCES .....	78
APPENDICES .....	85
APPENDIX I: The VARK Questionnaire .....	85
APPENDIX II: Source Codes .....	90

## List of Tables

Table 2.1: Comparison of e-learning models.....	27
Table 2. 2: Summary of most adopted learning styles.....	35
Table 2. 3: VARK model categories /dimensions with various teaching strategies.....	37
Table 4. 1: Summary of the various intelligent agents .....	53

## List of Figures

Figure 2.1: Reactive Architecture .....	13
Figure 2.2: Horizontal Layer Architecture.....	15
Figure 2.3: Vertical Layer Architecture.....	16
Figure 2.4: The phases of the Prometheus Methodology .....	20
Figure 2.5: An example of a deep neural network consisting of interconnected neurons .....	24
Figure 2.6: Personalized e-learning block .....	29
Figure 2.7: Online personalization Block.....	30
Figure 2.8: Offline Personalization Block .....	30
Figure 2.9: Main Components of Intelligent Learning System .....	33
Figure 2.10: Learning Management System Moodle .....	40
Figure 2.11: Learning Management System.....	41
Figure 2.12: Theoretical Framework Model.....	43
Figure 2.13: Conceptual Framework Model for the intelligent agent-based system.....	44
Figure 3.1: Prometheus Methodology Phases.....	46
Figure 3.2: Interaction diagram showing the behavior of the system.....	47
Figure 3.3: The three agents and how they interact with one another .....	48
Figure 4. 1: Machine learning process used to create the learner agent .....	54
Figure 4.2: Sample data from the JSON file.....	55
Figure 4.3: A simple tokenization.....	56
Figure 4.4: A neural network consisting of 8 fully connected neurons and two hidden layers....	61
Figure 4.5: Results of the neural network metrics used.....	62
Figure 4.6: System architecture .....	68
Figure 4.7: Notification of the learning style.....	71
Figure 4.8: Sample of the feedback/recommendation .....	72

## List of Abbreviations

AI	Artificial Intelligence
AUML	Agent Unified Modelling Language
BDI	Belief Desire-Intentions
HTML	Hyper Text Mark-up Language
ITS	Intelligent Tutoring System
LCMS	Learning Content Management System
LMS	Learning Management System
MaSE	Multi-agent Software Engineering
Moodle	Modular Object-Oriented Dynamic Learning Environment
PDT	Prometheus Design Tool
PVLE	Personalized Virtual Learning Environment
RMI	Remote Method Invocation
UML	Unified Modelling Language
VLE	Virtual Learning Environment

## **CHAPTER ONE**

### **INTRODUCTION AND BACKGROUND TO THE STUDY**

#### **1.0 Introduction**

This chapter covers the following subsections: background information of the study, statement of the problem, objectives of the study, research questions, and scope of the study.

#### **1.1 Background Information**

Teachers and students are increasingly turning to e-learning systems and applications as a result of technological improvements (El Fazazi et al., 2021). Despite the fact that each student has a unique learning style, taste, and area of interest, the traditional learning model often offers a wealth of educational materials to all learners (Hosni et al., 2020). There are many different ways for students to acquire information and knowledge (Balasubramanian & Margret Anouncia, 2018). While some learners prefer theories and mathematical models to grasp, others focus on data and algorithms, while others do better with verbal form and spoken explanations, and others do better with drawings, diagrams, and all other visual forms. Additionally, some students prefer to actively learn in groups, while others prefer to learn alone (Lakkah et al., 2017). Consequently, the outcomes of students' learning are significantly impacted by their characteristics. According to numerous studies, giving all students the same learning materials and teaching methods without taking into account their varied backgrounds, past knowledge, and learning objectives results in lower performance (Wu et al., 2018).

The teachers in a physical classroom should be aware of the preferences and learning preferences of the students they are teaching. They may find it very challenging to comprehend the various

students' learning styles. This is now achievable in virtual classrooms with adaptive e-learning systems thanks to technological advancements. For instance, employing technology for agents appears to be the primary strategy for resolving this issue. The employment of intelligent agents enables the development of a robust system that accommodates the demands and the learners' interests, giving the e-learning system adaptability and intelligence (Nadrljanski et al., 2018).

E-learning agents keep an eye on the online learning environment and enhance collaboration, which depends on the prior knowledge, social achievements, and learning preferences of the students. The e-learning agents also permit the study of new learning materials, allowing students to change the exhibited content to improve learning and teamwork outcomes in an e-learning environment (Fasihfar & Rokhsati, 2017). Personal learning, cooperative learning, and virtual learning are the three main e-learning methods. For personal learning, a major interest is chosen by the individual and looked at and evaluated through the internet and the individual asks the expert instructions some questions of their own indirectly. Concerning cooperative learning, an online discussion is crucial.

### **1.1.1 Components and Architecture of the E-Learning System**

A learning management system (LMS) and a learning content management system (LCMS) make up an e-learning system. An association's learning board is taken under the direction of the learning management system. The system serves as a turning point for many learning resources, and this tool programs the LMS and adds new capabilities available. Its attributes and qualities include individual guidance throughout the entire course, including the online classes, registration management, and data storage, concurrent management of numerous learning components, learning resource management and their presentations, access level management, and safety problems, saving progress, and performance management of students' interaction and

learning model systems. The LCMS is an information executives' framework that gives the chance to gather data in different structures and arrangements. The system assumes control over the administration of learning things accessible in learning storage facilities (Fasihfar & Rokhsati, 2017.)

Compared to traditional learning environments, many e-learning platforms fall short of providing excellent support. By enabling these settings to adapt based on the demands of the users, smart environments can assist in finding a solution. With the use of intelligent software agent technology, this is made possible since intelligent software agents can decide on their own without user input (Fasihfar & Rokhsati, 2017).

According to Dou and Ying (2012), the absence of customized studying is one of the shortcomings of traditional learning. E-learning has changed the conversational imparting method for tutoring that focuses on the instructors and puts more effort into learners' active learning in this way an individual gives more consideration to it. However, numerous instigations show that numerous e-learning systems need insight, which cannot give students direction in their studying as per their very own skills and conditions, and thus these systems cannot give the students personalized knowledge service. Additionally, these traditional systems have a few issues such as little interaction, unequal distribution of tutoring materials, and a lack of unique kinds of networks and study groups, which causes e-learning to appear untrustworthy to students (Duo & Ying, 2012). Along these lines, the utilization of related advances to take care of the current issues in e-learning turns out to be progressively pressing.

### **1.1.2 Personalized Learning Strategies**

Interaction and personalization are key characteristics of e-learning systems. The personalization feature contributes to the improvement of interactions in an e-learning system. Personalizing an

e-learning system aims at achieving customized learning through interaction with the learners. Some of the personalization techniques in the learning process include (Duo & Ying, 2012):

- i) customizing the user interface according to the different users such as tutors and learners
- ii) Customizing the study materials, for example, mixed media courseware, homework, and other individual data that are separated to students dependent on specific qualities of the channel rules. Tutors are able to recommend different learning materials to various learners according to their situations. The content of each learner is different from the other which fully incorporates personalized learning materials.
- iii) individualized learning exercises that are diverse intelligent and customized
- iv) Offering personalized guidelines whereby the system keeps the learning log of the learner which can then be analyzed to give individualized guidelines and recommendations to the learners
- v) customizing communication that involves the use of a group collaborative learning as a mode of learning and communication whereby close learners may choose to exchange information

Learners who apply Information and Communication Technology in their studying appreciate the open environment since they are in charge of their learning and are able to make choices of their own in the courses and the modules (Pour et al., 2017). Online educational tools have the content and also are able to interact with the learner depending on their approaches and level of understanding. This is possible through the use of intelligent agents. Intelligent agents participate in a crucial role in personalizing the e-learning environments. The agents offer the behavior of

the intelligent system and also cooperate to achieve the personalization of the e-learning environment.

## **1.2 Problem Statement**

Traditional e-learning systems lack intelligence which fails to give learners instructions in their learning according to their learning styles(El Fazazi et al., 2021), preferences, and interests (Hosni et al., 2020). Students have numerous approaches to learning and acquiring knowledge thus they cannot provide learners with personalized knowledge services (Balasubramanian & Margret Anuncia, 2018). In addition, they don't provide features to support a personalized learning approach and all students have access to the same activities and resources (Cakula & Sedleniece, 2013). While some learners do better with verbal forms and spoken explanations, others do better with drawings, diagrams, and all other visual forms. Additionally, some learners prefer to actively learn in groups, while others prefer to learn alone (Lakkah et al., 2017). The instructors in the physical classroom need to be aware of the preferences and learning preferences of the students they are teaching. They may find it very challenging to comprehend the various students' learning styles. This is now achievable in virtual classrooms because of the development of intelligent e-learning systems. For example, employing agent technology appears to be the primary strategy for resolving this issue. The employment of intelligent agents enables the development of a robust system that accommodates the demands and preferences of learners. Agents give the e-learning system adaptability and intelligence (Nadrljanski et al., 2018). In view of this, the study has utilized intelligent agents to accomplish personalized learning by identifying the learning style of the students using the VARK Learning style model. The agents were developed using the Prometheus methodology whereby these agents provide the learners with instructional resources that match their learning style.

### **1.3 Objectives of the study**

#### **1.3.1 Main objective**

This study aims to design and implement an intelligent system for supporting personalized e-learning.

#### **1.3.2 Specific objectives**

This study also aims to achieve the following objectives:

- i) To review the literature on how intelligent agents can support and improve personalized e-learning
- ii) To design a learner agent, tutor agent, and information agent that will assist in a personalized e-learning process based on a given learning style
- iii) To develop and implement a learner agent, tutor agent, and information agent that will assist in a personalized e-learning process based on a given learning style
- iv) To integrate the intelligent agents with an existing learning management system such as Moodle

### **1.4 Justification**

The study was important since was expected to offer a solution to college students with different backgrounds and learning styles through the development of an agent-based system. The intelligent agents will assist the students in their learning process which aims at improving the student's performance and also support the student-centric approach to learning.

### **1.5 Significance of the study**

This study was expected to be used by college students with the aim of supporting a student-centric learning approach as opposed to the tutor-centric approach as well as improving their

performance. The intelligent system can be used by learners in higher learning institutions to study as per their preferred method of learning. Through learning management systems like Moodle, the integration enables students to learn according to their preferred learning style. The study will also benefit the lecturers in that they won't have too much involved in the monitoring of student performance since the intelligent agents can monitor the student learning process and recommend the student the appropriate content based on the learning styles. Through learning management systems like Moodle, the integration enables students to learn according to their preferred learning style.

### **1.6 Scope of the study**

This study is meant to be used for higher learning institutions in Kenya to support dynamic learning (student-centric) and improve the student's performance. For demonstration purposes, one course (C programming course) was created in Moodle learning management system with four topics: arrays, datatypes, functions, and control structures

## CHAPTER TWO

### LITERATURE REVIEW

#### 2.0 Introduction

Traditional e-learning system lacks personalized learning and this is a weakness. E-learning does not involve the traditional way of teaching rather it emphasizes students' active learning. Various e-learning systems need insight that can't give students guidance in their picking up as per their very own conditions that are, the systems can't give the students the customized information administration that they require (Duo & Ying, 2012). Likewise, these systems additionally have a few issues, for example, frail connection, the uneven geographic scattering of showing assets, diverse structures of various types of systems, and learning gathering. This makes the students feel unconfident and suspicious of e-learning.

Traditional learning centers aim on transferring knowledge from tutors or experts to learners where the learner whereby the learner is a passive participant. These techniques are not sensitive to the learners' competencies, skills, attitudes, and educational background. In this, the interaction between the tutors and the learners is very limited (Moubaidin et al., 2013). The tutors deliver the content of the courses to the learners who rely on the tutors as a vital source of knowledge and information.

The internet has contributed a lot to making resources available for research and learning for both tutors and learners enabling them to share and acquire information (Abaidoo & Arkorful, 2015). E-learning uses the internet and other technologies to develop educational resources, instruct students, and control course content inside an institution. According to Abaidoo and Arkorful (2015), e-learning involves the use of information and communication technologies that

enables access to online learning materials (Abaidoo & Arkorful, 2015). The content in e-learning is delivered through electronic information and communication technologies

Hamidreza (2017) claims that while deploying intelligent agents, learning is tailored based on the individuals' prior data based on the student's prior knowledge, interpersonal highlights, and learning preferences, e-learning agents improve writing and learning efforts in the electronic learning environment. The agent should successfully provide new learning materials that allow students to change the displayed materials to improve the e-learning environment through learning and collaborative effort (Fasihfar & Rokhsati, 2017). In addition, a large number of e-learning environments do not provide adequate support in comparison to conventional learning environments thus smartizing these environments becomes very important. With the use of intelligent agents, this can be achieved such that a person can adjust to changes in regard to their personal needs (Fasihfar & Rokhsati, 2017). With the e-learning systems, the traditional way of teaching focuses on the tutor's changes and emphasizes the learner's active learning.

The rest of the chapter is organized into the following sections. Sections 2.1, 2.2, 2.3, and 2.4 review intelligent agents', intelligent architectures, numerous methodologies which are used in developing intelligent agents, and the AI techniques applied in intelligent systems respectively. Sections 2.5, 2.6, and 2.7 reviews personalized e-learning, e-learning systems based on intelligent agents, and various learning style models respectively. Section 2.8 reviews how personalized e-learning can be achieved through intelligent agents. Section 2.9 explores the Moodle learning management systems. Finally, section 2.10 reviews the theoretical framework.

## 2.1 Intelligent Agents

Creation of intelligent agents with intelligent behavior and the ability to behave like humans is made possible by artificial intelligence. Each intelligent agent can use sensors to analyze its environment and actuators to act on it (Russell & Norvig, 2021). There are various definitions of intelligent agents. An intelligent agent is a program that can complete routine and expected tasks (Russell & Norvig, 2020). Intelligent agents play a crucial part in e-learning. Many e-learning platforms may use improved support, therefore using intelligent agents to make them smarter could be a solution to this issue. (Fasihfar & Rokhsati, 2017). Intelligent agents help users communicate effectively with computer programs by making decisions automatically without requiring human involvement. Intelligent agents have various characteristics (Oskouei et al., 2014):

- (i) **Autonomous** - intelligent agents sense their surroundings and respond to them according to the rules set out by the designer and the information it perceives to have learned from it. This indicates that each agent is in charge of the work that is completed by themselves.
- (ii) **Learning/reasoning** - have the capacity to learn from their experiences and apply those lessons to change how they behave in various environments.
- (iii) **Reactive** - Every intelligent agent responds in accordance with the information they gather from their surroundings.
- (iv) **Goal-based** - Based on the knowledge they have about their surroundings and how best to accomplish those goals, intelligent agents have some objectives.
- (v) **Communication**- To accomplish their objectives, each agent must communicate with the environment, including other agents or users.

- (vi) **Cooperation** – To boost their abilities to accomplish their aims and complete the assignment quickly, they must collaborate with the other agents when working on complex tasks.
- (vii) **Mobile**- can use electronic communication networks for navigation.

Numerous researchers find agents as entities that act collectively with other agents thus a multi-agent system is used. A multi-agent system is made up of a number of autonomous entities that cooperate or compete with one another to achieve specific, individual, or collective goals. The main pros of multi-agent systems include: decentralization, are robust, extensibility, expertise, and common resources (Alexandru et al., 2015).

## **2.2 Intelligent Agents Architectures**

According to Wooldridge (2009), agent technology is a new paradigm that arranges and works in a dynamic and heterogeneous environment. The agent can monitor and respond to the changes proactively or reactively by itself by communicating with other agents to achieve certain goals or tasks on behalf of the user or other agents (Wooldridge, 2009). Building an agent-based system requires careful consideration of the agent architecture. In terms of reasoning and decision-making, resolving issues, and achieving objectives, it is comparable to an agent's brain (Chin et al., 2014). Techniques and methods for interacting with and supporting agents are included in agent architecture (Maes, 2018). An agent's architecture explains its components, capabilities, and how they work together (Bergenti et al., 2016). Common examples of typical agent architectures include logic-based architecture, BDI architecture, reactive architecture, hybrid architecture, cognitive architecture, and semantic architecture.

### **2.2.1 Logic-Based architecture**

This architecture, which uses symbolic representation for reasoning, is also known as symbolic-based/deliberative (Hopgood, 2016). The agent behavior is dependent on manipulating the symbols for representation since it models and expresses the agent behavior and the environment using symbols. The behavior of the agent, the goals generated and the actions that the agent can take are indicated by the specification of the agent (Russell & Norvig, 2021). The inference rules are included in the implementation to allow the agent to make decisions. Despite that the architecture is simple, it has the following weaknesses (Russell & Norvig, 2021).

- i) Symbol representation may not be accurate thus it may be difficult to model the environment,
- ii) Representing information in a symbolic form may also be difficult and thus may not allow the agent to think clearly in a time-constrained situation.
- iii) It might not be accurate to characterize the environment after converting the input from the percepts. Because the deduction process depends on the inference rules, it becomes highly challenging to put all the rules into practice for a situation the agent would face.

### **2.2.2 Reactive Architecture**

This architecture is dependent on the direct mapping of the situation to action. It varies from logic-based architecture since there is no central symbolic domain and symbolic reasoning is used. An agent reacts in a stimulus-response to changes in the environment. Input is transferred to the effectors in this architecture, which uses sensors and effectors to alter the environment. The Brooks Subsumption architecture is an illustration of reactive architecture (Chin et al., 2014). An example of a reactive architecture is shown in Figure 2.1, where a percept situation is translated into an action that reacts to the percept situation.

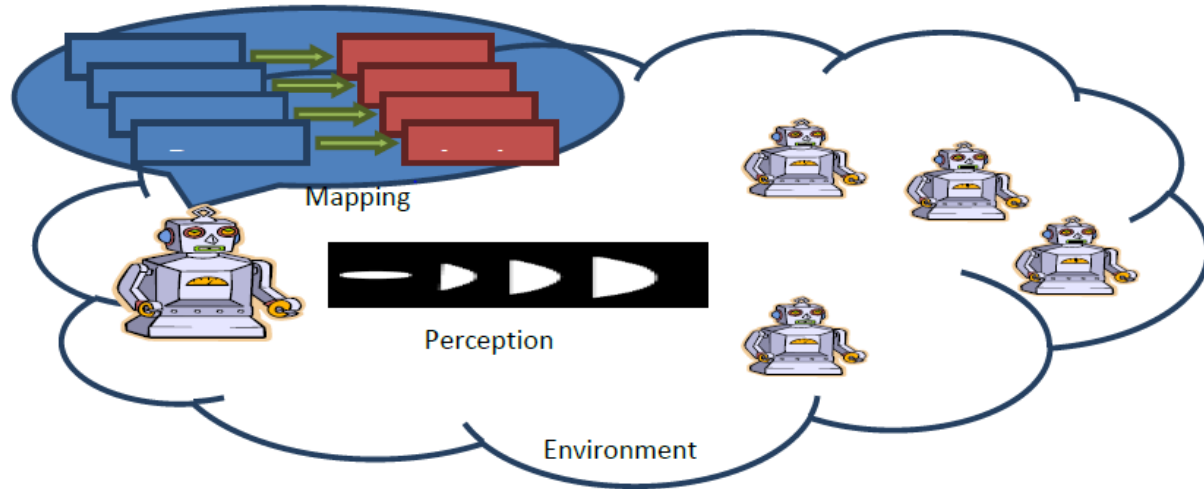


Figure 2.1: Reactive Architecture

Source (Chin et al., 2014)

This architecture is easy to design and implement than logic base architecture. However, the architecture has inadequate knowledge about the agent's present situation to determine the activation action for modeling the environment. Also, achieving learning in this architecture is difficult because processing the local information limits the planning capabilities in the future (Chin et al., 2014)

### 2.2.3 Belief-Desire-Intention (BDI) Architecture

This architecture, which is dependent on deliberative agent traits including belief, desire, and intention, is a mental state model. It uses beliefs and information to come to its findings. Beliefs, desires, and intents make up the three logic components that make up the agents in this architecture. The agent's knowledge of the world is made up of their beliefs. The desires are the motivating factor that enables the agent to accomplish its actions. The intentions are a commitment that the agent has toward their desires and beliefs (Chin et al., 2014).

The formal logic properties of this architecture are easy to learn because the agent subsystem's functional breakdown reveals the architecture's design. However, the efficient implementation of the functioning of the agent in the subsystem is not easy to understand. Thus, the agent needs to balance commitment and reconsideration. When the agent reconsiders it tries not to achieve an intention that may not be able to achieve. In addition, reconsidering too much may make the agent fail to accomplish goals because of inadequate time spent on the task. This architecture's design is logical and simple to comprehend. However, it is unclear how to implement agent functionality in the subsystem efficiently, therefore agents must strike a balance between commitment and reconsideration (Omicini, 2015).

#### **2.2.4 Layered (Hybrid) Architecture**

This design addresses the drawbacks of both the reactive and the deliberative systems while combining their benefits. In subsystems, various behaviors are addressed at hierarchical levels (Chin et al., 2014). Interactions both horizontally and vertically are conceivable. Each layer in the horizontal interaction has a direct connection to the sensory input and action output; as a result, each layer exhibits the behavior of an agent that maps an input to an intended action.

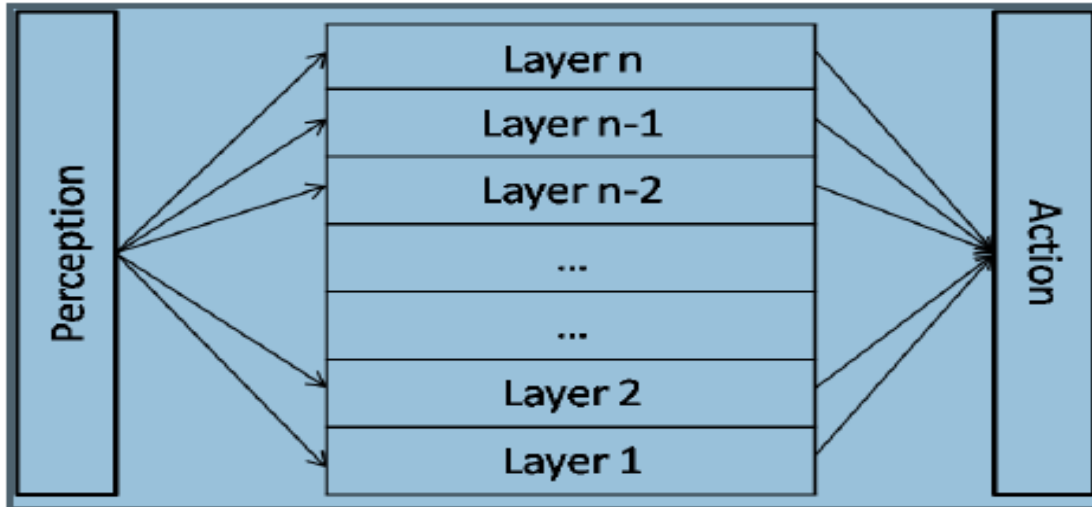


Figure 2.2: Horizontal Layer Architecture

Source (Chin et al., 2014)

In a horizontal layer design, just  $n$  layers are needed to map  $n$  separate behaviors. But a mediator function is necessary because layer interactions take inconsistent actions. The vast array of potential interactions between horizontal layers is still another difficulty (where  $m$  represents the number of actions per layer). By having sensory input and action output that are individually handled by no more than one layer in each, the vertical layer interaction offers a solution to the issue in the horizontal layer interaction. In this interaction, control passes from the bottom layer, which generates the action output, to the top layer, which collects data from sensors.

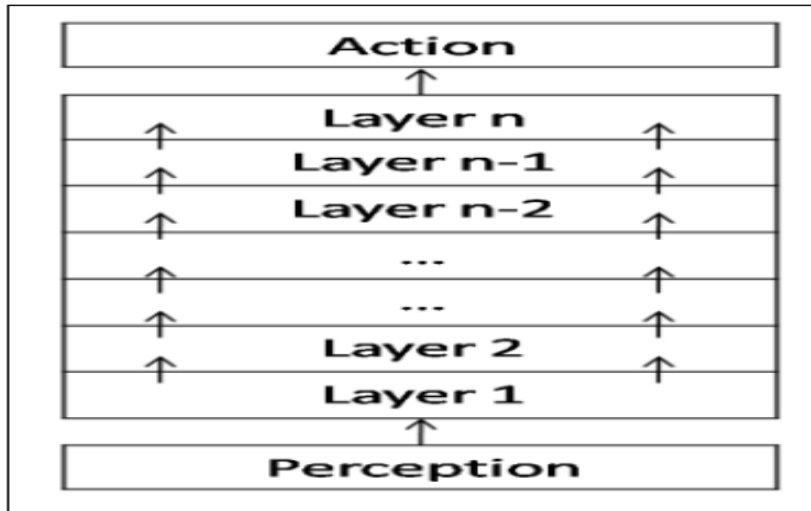


Figure 2. 3: Vertical Layer Architecture

Source (Chin et al., 2014)

Despite its advantages, its main drawback is that the interaction between the layers is reduced to  $m^2(n-1)$ . Also, in case one layer fails, the whole system goes down (Chin et al., 2014).

### 2.2.5 Cognitive Architecture

This architecture makes use of cognitive sciences, which are concerned with how people think and feel. It began with architecture known as production systems and advanced with time (Chin et al., 2014). It differs from a multi-agent approach in the following ways (Emelyanov et al., 2016).

- i) It has a formal way of encoding knowledge and links it with its interpreter
- ii) Representing the procedures that impact knowledge is accompanied by strong assumptions
- iii) It presumes a modular way of representing knowledge

- iv) It provides an intelligent behavior system level instead of a component level which is designed to provide special duties
- v) It provides an integrated approach whereby the necessity for such precise craftsmanship is reduced by a shared set of representations and mechanisms.

## **2.3 Methodologies for Developing Intelligent Agents**

A methodology is an organized, speculative analysis of tactics related to a topic of study. It contains concepts like worldview, fictitious model, stages, and objective or quantitative procedures. A methodology instead provides hypothetical support for understanding whether technique, collection of strategies, or best practices can be tied to a certain scenario rather than setting out to provide arrangements. Some of the existing methodologies for developing intelligent agents include Prometheus methodology, MaSE methodology, and Tropos methodology among others as discussed below:

### **2.3.1 The Prometheus Methodology**

This methodology comprises three phases: the system specification phase which focuses on the identification of the system's fundamental operations and the inputs (percepts), outputs (actions), and any other shared data sources that are crucial. The architectural design phase involves the identification of agent types and how the agents interact. The detailed design involves the development and definition of each agent's internal structure in terms of capabilities, data, events, and plans (Lhafiane et al., 2015). The phases are discussed below:

#### **2.3.1.1 System Specification Phase**

This phase attempts to specify the interface between the system and its environment in terms of actions and perceptions, establish the system goals, construct use case scenarios, explain system

operation, and identify the fundamental functionalities (Lhafiane et al., 2015). Defining the functionality of the system is important since this defines the information that the agent requires and also the information it produces. The use case scenarios in this methodology depict the steps involved in the system in operation. Each step involves an explanation of the functionality which is in charge of cross-checking for consistency using functionality descriptors (Shehory & Sturm, 2016). The following are the perspectives developed in this phase:

- i) Defining the system's objectives, sub-objectives, and associated descriptors
- ii) Creating scenarios that adequately address the objectives and can offer a process-oriented perspective on the system to be created.
- iii) Outlining the features connected to at least one objective and seeing how the system behaves
- iv) Describe the relationship between the agent system and the setting in which it is placed.

### **2.3.1.2 Architectural Design Phase**

This phase uses the system specification artifacts to build the system architecture which involves three steps: firstly, identification of the agent types, secondly, specifying the interaction between the agents identified and lastly designing system overview diagrams (Lhafiane et al., 2015)

The agent types are formed by combining functionalities. The functionalities are allocated to the agents by breaking down the artifacts of the previous stages to advocate the possible assigning of the functionalities of the agents. These functionalities are assessed by traditional software engineering using the concepts of cohesion and coupling. This concept builds up the expected agent types and the scenario descriptors from the specification phase (Padgham & Winikoff,

2012). The improvement of the determination of the interaction between the agents has the following stages:

- i) Utilizing use case scenarios to create the interaction diagrams
- ii) Using interaction protocols as a generalization of interaction diagrams
- iii) Creating message and protocol descriptors

Consistency checking is performed to check whether the agent's descriptors are steady with the arrangement of functionality descriptors and this is possible with the help of the Prometheus Design Tool (PDT). Interaction diagrams are used to indicate the interactivity between the agents. The use cases that are developed in the specification stage are used to come up with the corresponding interaction diagrams thus consistency checking is done (Manate et al., 2014).

### **2.3.1.3 Detailed Design Phase**

This phase encompasses the definition of the agents' internals and the process diagrams (Manate et al., 2014). It uses the system architecture artifacts and it focuses on the definition of the capabilities, internal events, plans, and detailed data structures (Lhafiane et al., 2015). The following perspectives are addressed in this phase:

- i) Deciding the capabilities that the agent needs to fulfill
- ii) Using an agent overview diagram to explain how the capabilities relate to one another
- iii) Creating protocol standards that outline an understanding of the internal processing of specific agents
- iv) An overview of the information in the capability descriptors for each capability

The agent overview diagram's objective is to present a high-level picture of the internal workings of the agent within the system. Individual plans, events, and data descriptors are among the

artifacts in this phase that offer the information needed to move forward with the system implementation (Padgham & Winikoff, 2012).

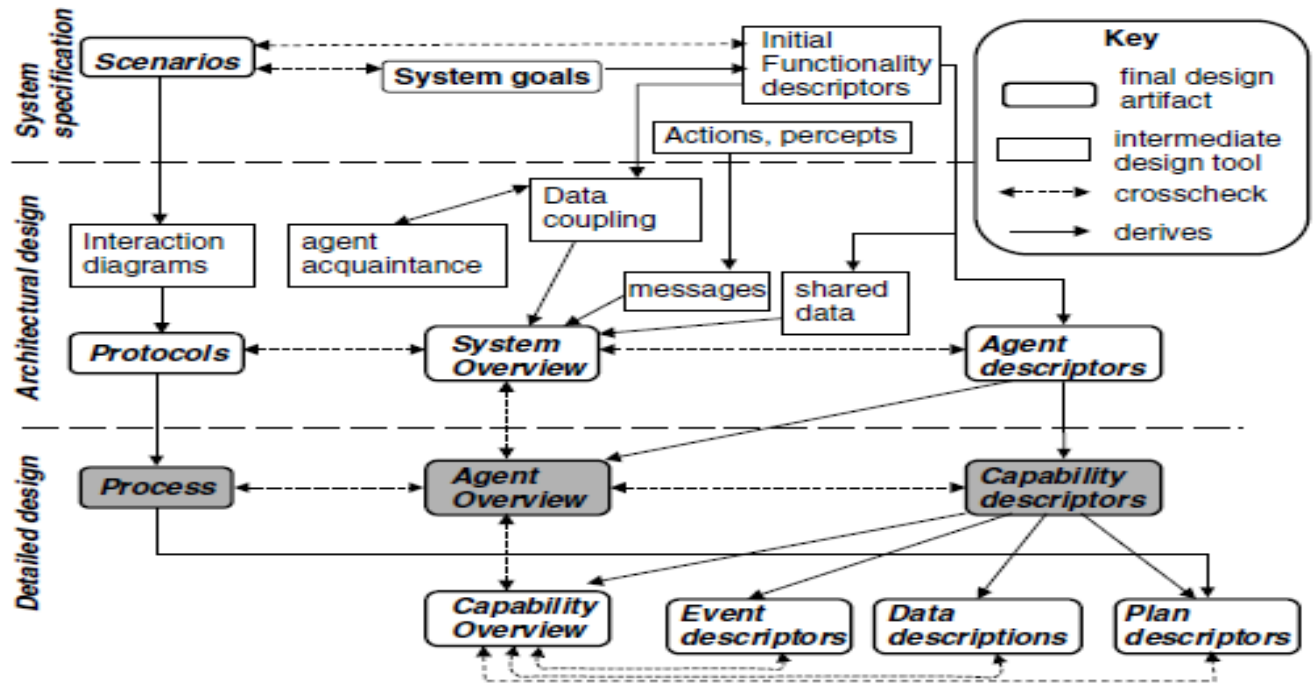


Figure 2.4: The phases of the Prometheus Methodology

Source (Lhafiane et al., 2015)

### 2.3.2 MaSE Methodology

The objective of this methodology is to offer an entire lifecycle technique to assist system developers in creating a multi-agent system agent. It completely depicts the procedure that directs the development of a system from its specification to its implementation (Shehory & Sturm, 2016). It has two phases (the analysis phase and the design phase) which have seven steps. The phases are discussed below:

### **2.3.2.1 The Analysis Phase**

This phase consists of three steps. First, the goals are captured and this involves the identification of goals and structures. Second, use cases are applied which involves the extraction of main scenarios from the initial system context or duplication of them if they exist. These use cases assist in the development of sequence diagrams. Third, roles are refined which involves the construction of a role model and a concurrent task model. The former describes the roles in the system while the latter represents the task graphically (Shehory & Sturm, 2016).

### **2.3.2.2 The Design Phase**

This phase consists of four steps. First, agent classes are created whereby an agent diagram is an output describing the entire multi-agent system. The agent and the roles they play are shown in this diagram. The conversations available are indicated by links in the diagrams and are labeled with a conversation name. Second, the conversation is constructed using a communication diagram. Third, agent classes are assembled which involves the definition of the agent architecture and its components. This methodology does not have a specific implementation platform when it comes to its architecture. The final step involves designing the system whereby a deployment diagram is built specifying the locations of the agents within the system. This methodology has an automated agent tool that implements all the seven steps which transform the analysis model into design constructs (Shehory & Sturm, 2016).

### **2.3.3 Tropos Methodology**

This methodology consists of the following phases: early requirements, late Requirements, architectural design, detailed design, and implementation as discussed below.

### **2.3.3.1 Early Requirements Phase**

The plans of the stakeholders are the main focus of this phase and are transformed into goals. The system to be designed must meet both functional and non-functional criteria, which are determined by modeling these initial goals. In this stage, goals are represented by actor and goal diagrams. The actor diagram depicts the domain's stakeholders and their connection, showing how the actors rely on one another to accomplish tasks, carry out plans, and provide resources. The goal diagram shows how the objectives and strategies of a particular actor who is responsible have been analyzed. Goal and plan analysis depends on reasoning techniques like contribution analysis, AND/OR decomposition, and means-end analysis. This method aids the analyst in organizing the system goals, finding soft goals, offering strategies and resources designed to achieve a goal, and identifying objectives that support or obstruct the accomplishment of other objectives (Morandini et al., 2014).

### **2.3.3.2 Late Requirements Phase**

This phase involves the extension of the models that are created in the early requirement phase. This phase is significant because the target system is modeled in its environment. The interdependence with the other actors in the models contributes to the achievement of the stakeholders' goals thus, these dependencies define the target systems' functional and non-functional requirements whereby the goals are further divided into sub-goals (Morandini et al., 2014).

### **2.3.3.3 Architectural Design phase**

This phase involves the inclusion of new actors which are described by an extended actor diagram. The derivation of the new actors is based on the choice of architectural design. The new

actors are meant to fulfill non-functional requirements or to support the sub-goals from the late requirement phase (Morandini et al., 2014).

#### **2.3.3.4 The Detailed Design phase**

This phase involves the definition of the specification of the agents at the micro-level. The capabilities of the agent are depicted using UML activity diagrams, the plans of the agents are represented by the plan diagrams, and the interaction between the agents is represented by agent interaction diagrams which are the AUML interaction diagrams (Morandini et al., 2014)

#### **2.3.3.5 The Implementation phase**

This methodology implements the agents using the BDI platform, in particular the JACK intelligent agent. Agents, capabilities, database relations, events, and plans are the five primary language constructs used in JACK. The only thing system developers need to do is map each design notion to one of JACK's five constructions. Several suggestions and heuristics are also provided by this methodology for mapping Tropos notions to BDI concepts and BDI concepts to JACK components (Morandini et al., 2014).

### **2.4 AI Techniques Applied in Intelligent Agent Systems**

#### **Deep Neural Network**

Deep neural networks and other machine learning techniques are essential tools for a variety of applications, including image processing, audio recognition, and natural language processing. These methods typically produce predictions with a high level of precision, often on par with human ability. A deep neural network is a group of neurons arranged in a series of numerous layers, where the neurons take the activations of the neurons in the layer above as input and conduct a straightforward calculation, such as a weighted sum of the input followed by a

nonlinear activation. Together, the network's neurons create a sophisticated nonlinear mapping from the input to the output. Error propagation is a method for adjusting each neuron's weights that are used to learn this mapping from the data. Figure 2.5 shows an example of a deep neural network

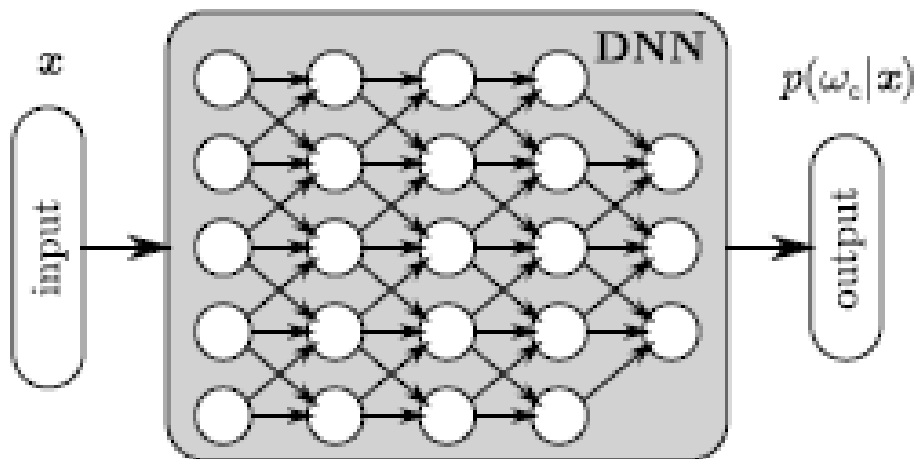


Figure 2.5: An example of a deep neural network consisting of numerous interconnected neurons  
Source (Montavon et al., 2018)

Interpretation of the concept is a must and usually, it is represented by the neuron in the top layer. The neuron at the top is abstract meaning that one can see it, meanwhile, the input domain of the deep neural network can easily be interpreted. When developing a prototype it is important to formulate the activation maximization framework.

### Activation Functions and Activation Maximization

The activation functions are used in determining if the output of the neural network is a yes or no. It does so by mapping the output values between 0 to 1 and -1 to 1 depending on the function

used. The function is involved in deciding whether a neuron should be activated or not through the calculation of the weighted sum and further adding bias to it. The main purpose of the activation function is introducing non-linearity into the output of the neuron. Non-linear activation functions are needed because without the activation the neural network would just be a linear regression model. Various activation functions include the sigmoid function, softmax function, ReLu (Rectified linear unit), Tanh function, and linear function. The sigmoid function has a range value of (0, 1) compared to the linear functions which have a range of  $(-\infty, \infty)$ , and thus the range is bounded and therefore blowing up during the activations is simple to avoid. It has been around for a very long time and is the most popular activation. The Tahn function has an output range of (-1, 1) which is an extension of the sigmoid function. The negative inputs of this function are mapped to negative output values as well as the input values which are near zero are mapped to output values that are near to zero therefore the network is not stuck during training. The Tahn function is preferred to the sigmoid as the derivatives it is large compared to the derivatives of the sigmoid near zero (Szandafa, 2021).

Our approach adopted the softmax activation function which is a combination of multiple sigmoid functions having a range of values (0,1). The range is treated as a probability of a particular class of data points. The function for every data point of all the individual classes returns the probability of the data points.

## **2.5 Personalized e-Learning**

Customizing e-learning to meet the unique demands of the learners is known as personalized e-learning. This customized e-learning is delivered by figuring out learners' entry-level options, maintaining the students' numerous learning preferences, and modifying the learning strategy, which has a variety of choices such as role selection, which lets the learner choose the

appropriate path rather than having to read pre-assessments for all of the content, which are based on the learner's achievement, and lastly through surveys, which let a student select their preferred method of learning (Pandey, 2017). With personalized eLearning, students can set their objectives, create workable achievements, choose their study path, learn on their own time, choose the degree of engagement they consider appropriate, receive an individual response, and use the advice provided to them to further their education (Pandey, 2017)

### **2.5.1 Prior Studies on Personalized e-Learning**

E-learning is moving from being teacher-driven to student-driven with the goal that personalization (picking up as indicated by the person's advantage, information base, and style), and gaining adaptability (time and area) are improved. A huge work in the instructional plan of current Virtual Learning Environments is grounded in objectivism. The instructor is the point of convergence of control of the learning material, with the assumption that the instructional procedure is predictable (inferring a biased structure and succession for guidance). Virtual Learning Environments created under objectivism support knowledge exchange from tutors to learners. In any case, VLEs empower learning sufficiency when they change in accordance with the requirements of individual students (Xu et al., 2014).

The constructivist learning paradigm serves as the foundation for personalization. Traditional education assumes that teachers will deliver instruction and that learning will take place in a classroom through natural and incisive teaching techniques that organize the learning environment. Personalization in VLEs assumes that every online student is a unique individual with an apparent learning style, pace, and method. These can be viewed as student-focused, two-way intuitive, and dynamic learning procedures of information development. They give applicable appropriate toolsets by empowering a person's picking up, bringing about a model

where the student needs to drive the learning procedure. PVLEs give customized e-learning circumstances to online understudies to heighten and expand abstract capacities similarly to form their one-of-a-kind thinking methods. Table 2.1 presents a comparison of PVLE components under both constructivism and objectivism and their related frameworks usage contemplations (Xu et al., 2014).

Table 2.1: Comparison of e-learning models

Learning Process	Constructivism (learner –centered)		Objectivism (instructor-entered)	
	Pedagogical Model	System Implementation	Pedagogical Model	System Implementation
Stage 1: Learning	<ul style="list-style-type: none"> <li>• Content modeled via situations rather than knowledge structures</li> <li>• Concern with processes whose perspective and interpretations can be constructed</li> <li>• Should be in realistic settings</li> </ul>	<ul style="list-style-type: none"> <li>• Learning materials are structured.</li> <li>• Each concept linked by keywords, consequence and related issues</li> </ul>	<ul style="list-style-type: none"> <li>• Content presentation modeled in terms of structures</li> <li>The goal of teaching is to transfer knowledge, thus presentation is critical</li> </ul>	<ul style="list-style-type: none"> <li>• Learning materials are predefined by instructor via electronic documents.</li> <li>• Learning materials controlled by instructor</li> <li>• All receive the same material</li> <li>• Instructor initialed approach</li> </ul>
Stage 2: Interaction	Instructors provide feedback on process <ul style="list-style-type: none"> <li>• Feedback mediates understanding and mastery of topic</li> </ul>	Two-way interaction. System provides analytical feedback on student's performance. Student issues questions along with his/her study <ul style="list-style-type: none"> <li>• Dynamic approach</li> </ul>	Interaction occurs via electronic communication. Students evaluate learning progress and needs, complementing the high degree of learning control	One-way interaction <ul style="list-style-type: none"> <li>• System provides static feedback on performance based on pre-defined answer, narrow approach</li> </ul>
Stage 3: Self Evaluation	<ul style="list-style-type: none"> <li>• Self-evaluation test integrated with the task and not separate activity</li> <li>• Students actively participate and interact through quiz routines: self-testing and mastery of learning</li> </ul>	Self-test integrated with learning <ul style="list-style-type: none"> <li>• Test generated dynamically based on student's progress</li> <li>• System diagnoses self-test results</li> </ul>	Self-evaluation is student's reflection on the results of testing of what has been learned	<ul style="list-style-type: none"> <li>• Self-evaluation test integrated with learning</li> <li>• Test pre-defined for all students</li> <li>• System provides simple right/wrong answers without analytical analysis</li> </ul>

Source((Xu et al., 2014)

Research demonstrates that each learner receives educational materials dependent on their exceptional learning style, needs, and interests (Cakula & Sedleniece, 2013). There is a trend to make personalization an inherent component of online learning. An e-learning platform's content is accessible from anywhere at any time, and it appears to meet each student's unique needs. However, simple access to educational resources improves instructing and learning outcomes (Zajac, 2009). E-learning platforms are already widely used, however, they do not now allow elements that would allow for a tailored learning strategy, and all students may easily access comparable resources and activities. Personalization entails modifying the learning backgrounds of distinct students by considering their knowledge, skills, and learning preferences (Bhansali, 2016). Time, space, and other restrictions on the showing process are removed with personalized learning, which also plans to adapt instruction to each student's continuously evolving demands and abilities (Howe & Kekwaletswe, 2010). A student's personality - learning styles, the structure of information - the semantic web, and the technology approach are three primary paths of development that the authors identify while studying the topic of customization in e-learning. According to Cakula & Sedleniece 2013, personalized e-learning has four blocks as in figure 2.6 below

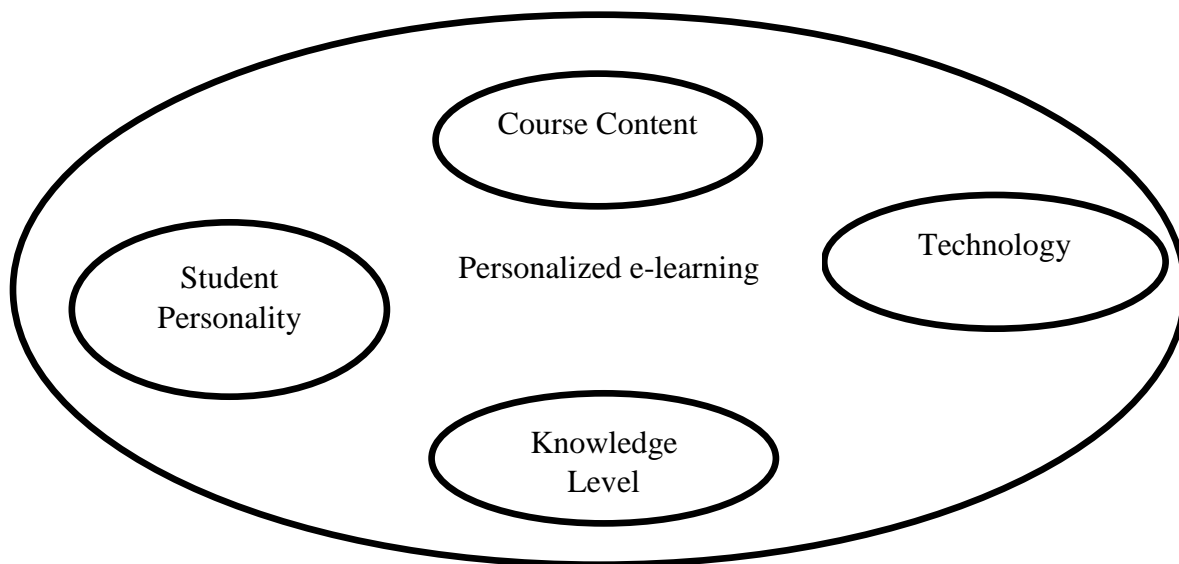


Figure 2.6: Personalized e-learning block

Source((Cakula & Sedleniece, 2013)

### **2.5.2 Personalized e-Learning and Content**

Information and Communication Technology (ICT) supports and encourages electronic learning, also known as e-learning (Tengah, 2009). Depending on the intuitiveness that is stressed, two categories of successfully developed e-learning systems can be identified classroom (El Mhouthi et al., 2017):

- i) **Static online education** in which students merely download the necessary instructional resources. The content administrator can simply upload files, such as HTML, PowerPoint, PDF, or video documents. This system does not encourage intelligent correspondence and intuitive learning conditions. Regardless, this system is as yet helpful for students who gain freely from different content posted in the framework.
- ii) **Dynamic e-learning-** in this, there are various varieties in this classification of system. They include, for instance, discussion forums, chatting, email, e-content management, and evaluation techniques. Learners can adapt freely outside of the customary study classroom with a similar environment in contrast to the physical

According to Gomez and his colleagues (2008), depending on their preferences and usage level, students can demonstrate unique subjective learning styles and increase their proficiency with the e-learning system. There are two sorts of personalization: online personalization and offline personalization. Figure 2.7 shows online personalization by observing student connections in the

system ceaselessly and continuously, and as needs are given the proper content (Gomes et al., 2008).

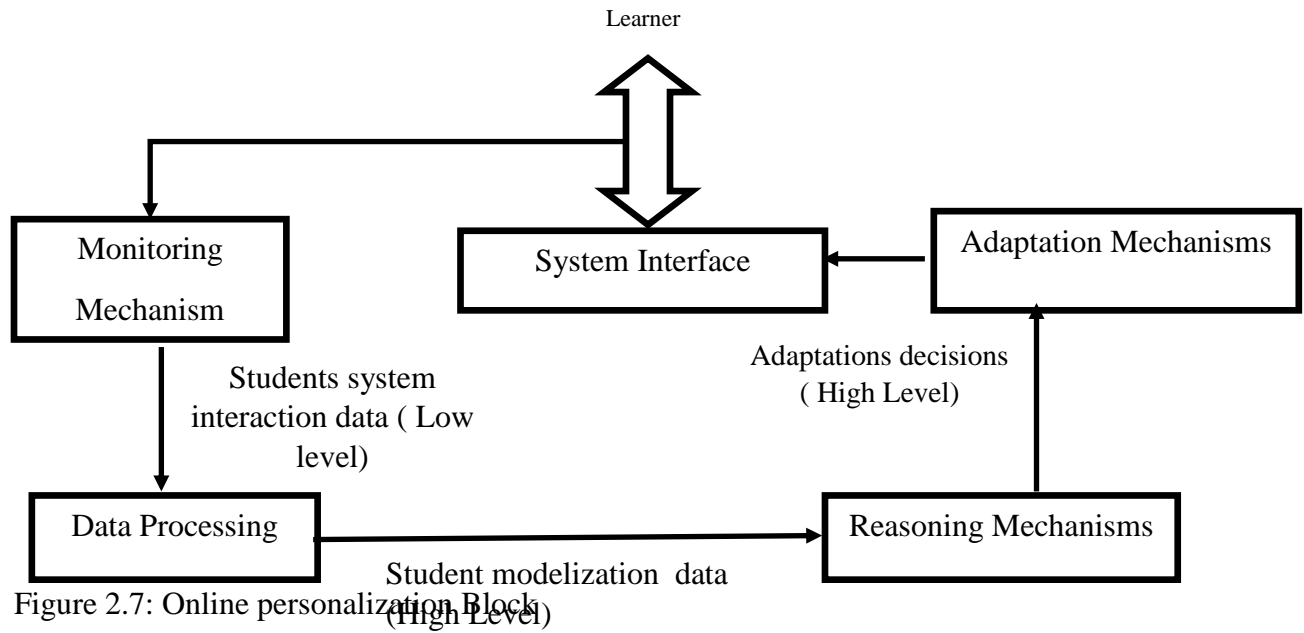


Figure 2.7: Online personalization Block  
Source (Tengah, 2009)

Figure 2.8 illustrates how offline personalization data from students are combined and processed to provide changes to the course content.

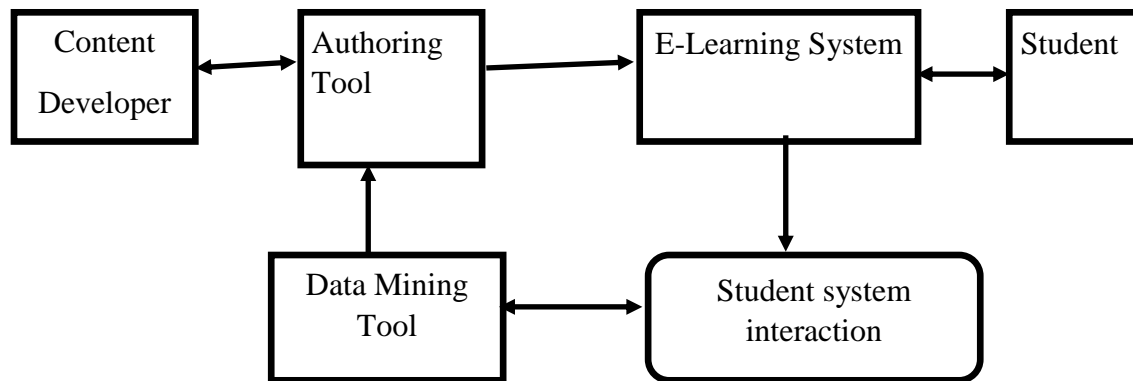


Figure 2.8: Offline Personalization Block

Source (Tengah, 2009)

## **2.6 Overview of e-Learning Systems Based on Intelligent Agents**

### **2.6.1 Intelligent Tutoring System**

Intelligent tutoring systems (ITSs) are computer-based educational tools that use training methods to show what should be taught and models of instructional content to determine what should be taught. They have an impact on inferences about students' understanding of subjects or assignments so that the guidance's manner or content can be gradually changed. The content model gives the ITS depth with the goal that learners can learn by doing in practical and important settings. The models take into account the content to be made on the fly. It likewise permits a blended initiative tutorial exercise connection where learners can make inquiries and have more authority over their learning. There are instructional models that permit the computer tutor to intently approach the advantages of individualized guidance by computer instructors (Dermeval et al., 2018).

These systems empower searching models of learners' dimensions of learning and use learning procedures to build learners' information of data. They depend on the improvement and execution of strategies and systems of artificial intelligence and on this premise the content and the strategy for training presentation of topics can change in accordance with the individual capacities of the learner. They are planned to help and improve the instructing and learning process in the chosen field while regarding the uniqueness of the learners (Dermeval et al., 2018).

Traditional intelligent tutoring systems are based on the region advantage they are imagined to compensate and get ready appropriately, their organization systems are much the time space-based more cutting edge ITSs give extra thought to unquestionably gotten issues and thoughts of

the tutoring approach. An intelligent tutoring system depends on information since they have (Saulnier et al., 2008):

- i) knowledge about space learning,
- ii) information on the guideline by which students are instructed, and
- iii) strategies by which to apply the standards and information of strategies and systems for demonstrating the progression of students to get learning and aptitudes

### **2.6.2 An Agent-Based Intelligent Tutoring System**

The agent-based intelligent tutoring system is a highly reusable intelligent tutoring system that can broaden a conventional course management system with a lot of insightful capacities that permit both learner modeling and automatic curriculum generation. The addition of an Agent-Based Intelligent Tutoring System permits any customized electronic technology to be adaptive. The main prerequisite of the course management system is that it can stretch out with scripting language supporting RMI invocation and have the capacity to get to outside information sources (Marciniak, 2014). This system supports automatic curriculum generation to such an extent that every learner is allocated to at least one unique course. Rather than utilizing an arranged rundown of learning objects, a curriculum is utilized to give to a particular student all the essential information to finish a particular course. While learning objectives show what the student needs to learn, the curriculum determines how the ideas can be learned.

The advancement of an intelligent agent in a customized e-learning system assumes a significant job in improving e-learning. An intelligent agent has the capability of doing a task for other people thus it is able to do an analysis of the profile, the quality of the information, and, the learning limit in a personalized e-learning system. The intelligent agent installed in the e-learning system breaks down accessible teaching models. It is also able to actualize a learning

methodology educationally by clarifying the content positions ranks of feedback received, and how the materials are instructed or clarified. The agent arranges learning materials from Web 2.0 technology in the information archive and representation within the created system is dependent on the ontology of learners and tutors (Tengah, 2009).

To achieve personalization for e-learning clients, clients alter the e-learning interface for the resources they have accumulated themselves and the resources that have been suggested by the system. An intelligent agent composes the learning materials by methods of ontology to represent the knowledge. As shown in figure 2.9, the built e-learning system implements the five core concepts of the Intelligent Learning System: student model, pedagogical module, communication model, domain knowledge, and expert model.

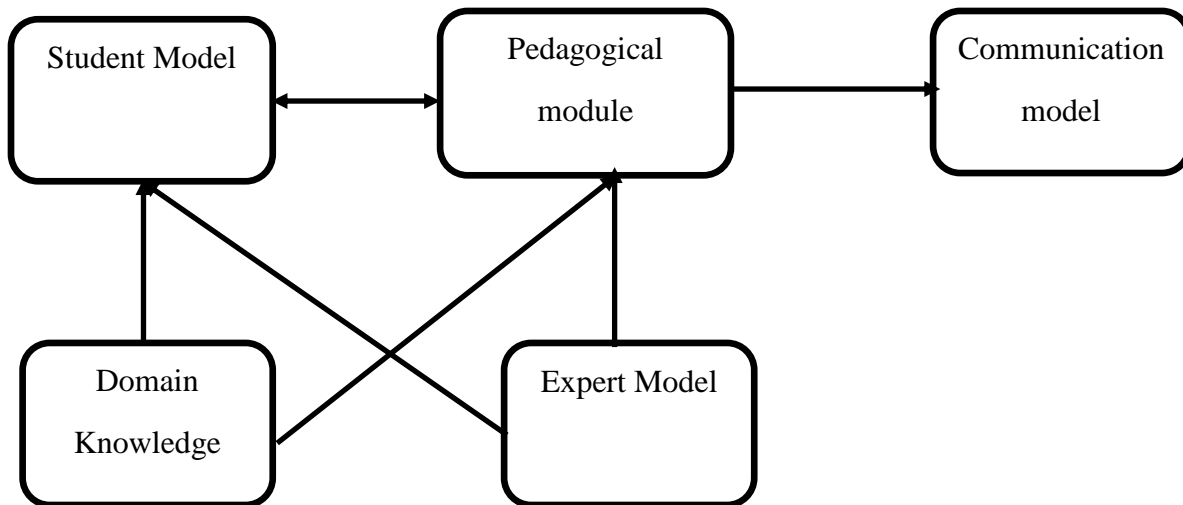


Figure 2.9: Main Components of Intelligent Learning System

Source (Tengah, 2009).

### **2.6.3 Personalized Intelligent Multi-Agent Learning System**

According to Bokhari and Ahmad (2014), multi-agent-based systems help in the development of interactive and better e-learning systems (Bokhari & Ahmad, 2014). The system is based on the learners' learning styles as well as personal characteristics using the semantic web and ontologies. Felder Silverman's style model is used to identify the learners' learning styles, after which comprehensive open dynamic learners' profiles are developed and evaluated using student assessment approaches. After the learners' profiles are established, an ontology-based personalized recommender system is designed to recommend appropriate learning components to a certain learner according to their profiles. In accordance with students' preferred learning components, the recommender system creates a preference list. A multi-agent approach is used to build the personalized intelligent system which consists of various intelligent software agents including; a software agent that identifies the learning styles, a software agent that creates the learner profile, a pedagogical suitability software agent, and lastly, a software agent that creates optimal learning scenarios (Melesko & Kurilovas, 2017).

### **2.7 Overview of the Different Learning Style models**

Many scholars have put forth various definitions of learning styles. As an illustration, Chick (2016) described learning style as the process through which learners receive, sort through, interpret, organize, draw conclusions, and store information for later use. According to Jaleel & Thomas (2019), learning style refers to the idea that everyone has a different preferred method of instruction or study. Learning style, as per Kumar et al. (2017), is a learner's preferred method of instruction. They said that it was characterized by the characteristics, traits, and interests that people learn, retain, and use the information.

There are many learning style models available that pedagogical specialists employ to provide tailored e-learning. Over seventy learning style models exist but the most adopted are the Felder-silver model (Zagulova et al., 2019), Kolb model (McLeod, 2017), Honey- Mumford (Kumar et al., 2017), Myers-Briggs Type Indicator (Imran Hussain, 2019), and the VARK model (Kumar et al., 2017) as discussed in Table 2.2.

In this study, VARK model was used to show how a student's learning style can be determined and how to utilize intelligent agents to deliver learning resources that are suitable for the student's learning style. For a number of reasons, the study employed the VARK model as the foundation for personalized e-learning. It is acceptable to experts and suitable for the majority of educational systems. Additionally, it is incredibly user-friendly and the results are quite simple to interpret (Daoruang et al., 2018). Also, it largely depends on a person's personality type (Kumar et al., 2017). This aligns with this study as the interest is the learning characteristics (learning styles) of the students. Table 2.3 lists the dimensions/categories of the VARK model.

Table 2. 2: Summary of most adopted learning styles

<b>Learning Style Model</b>	<b>Learning Style Dimensions</b>	<b>Description</b>	<b>Learning Style Instrument</b>
Felder Silverman Model	Active /reflective	<b>Active students</b> are more likely to prefer working in groups and interacting with one another. <b>Reflective learners</b> enjoy deliberating and considering their thoughts. They choose to work independently and are not motivated to study alongside the other students.	Index of Learning Styles (ILS)
	Visual /Verbal	Illustrations, graphs, photos, and diagrams, for instance, help <b>visual learners</b> remember information the best. They employ the concept of mapping learning and frequently make dubious coding comments. A <b>verbal learner</b> takes in more information from	

		words and discusses how to better understand pictures and graphs. taking notes while listening to their classmates	
	Sequential/ Global	To uncover arrangements, <b>sequential learners</b> learn in straight lines and at regular, slow incremental pauses. <b>Global learners</b> are reliant on integrated thinking and move at rapid speeds. They favor unconventional learning materials and learn the most effective ways to resolve challenging issues as a result.	
	Sensory /Intuitive	Learning takes place via sensory or visual thinking, with an emphasis on facts and concepts, as opposed to abstract thinking, with an emphasis on theory and beyond.	
Kolb model	Converging	A student often relies on comprehension of theory and active experimentation	Learning Style Inventory (LSI) Revised Inventory
	Diverging	Learners put more emphasis on practical experimentation and critical judgment	
	Accommodating	Students emphasize practical exploration and engaged experimentation. prefer to complete tasks, plans, and new ideas, and are excellent at responding to changes in situations and solving difficulties instinctively and experimentally	
	assimilating	Reflective perception and abstract conceptualization are preferred by learners. They enjoy making assumptions and formulating theories; they are more interested in thoughts and theoretical concepts than in people, and they think that ideas are more useful when they are intelligently grounded.	
Honey & Mumford Model	Activist	They acquire knowledge by experience, and they typically employ an open-minded approach by choosing a variety of tasks to gain knowledge.	LS Questionnaire (LSQ)
	Reflector	They pick up knowledge spontaneously and through seeing examples. They favor doing analyses, reports, and learning from fresh experiences.	
	Theorist	They acquire knowledge and comprehend learning materials according to their environment, including models, hypotheses, and certainties, all the while keeping their objectives in mind.	
	Pragmatist	They seek out ways to put what they have learned into practice and favor doing experiments to test out fresh suggestions and theories from professionals.	
Myers Briggs Types Indicator	Perceive/judge	Learners who judge are certain, organized, and self-controlled. They put more effort into finishing their jobs and responsibilities. Learning through perception is engaging, adaptable, and unrestricted. They prefer to start with assignments	Myers Briggs Types Indicator (MBTI)

		so they can fully understand their project.	
	Sense/intuitive	Those who learn intuitively discover links and examples between the facts they have gathered. Sensitive learners prefer to receive information in fictional universes	
	Think/feel	They concentrate on how people respond to circumstances as well as how they interact with the outside environment.	
	Extraversion/introversion	Learners that are introverted concentrate on concepts, ideas, and abstractions Learners that are extraverted favor taking action and communicating with others.	
VARK( Visual, Auditory, Read/Write, Kinesthetic )	Visual	Visual learners tend to employ graphs, pictures, hierarchical models, and symbolic tools like arrows that express written information.	VARK Questionnaire
	Auditory	Instead of taking notes during lectures, students would rather listen and converse with other students on the material being covered in class.	
	Read/write	When learning, students prefer to use printed words and text.	
	Kinesthetic	Learning is best done by doing. They tend to prefer hands-on learning to watching or listening, and they typically perform poorly in the classroom.	

Table 2. 3: VARK model categories /dimensions with various teaching strategies

Learning style dimension	Teaching strategy	Learning object
Visual	Images/diagrams, charts, slides posters, videos, posters, graphs, mind maps	Video PowerPoint slides
Aural/auditory	Topic discussion, talk thoughts, remembering stories, recording notes	Presentation slides with audio, audio videos, and recorded notes
Read/write	Headings and Lists Written notes Definitions and text manuals	PowerPoint slides Text documents
Kinesthetic	Doing practice tests Experimental methods	Practicals Hand on exercise

## **2.8 Personalized E-Learning and Intelligent Agents**

Getting information from the web has been very essential in the learning process for a while, now, however, nowadays it is designated as a very important source of information. Learning helps with dealing with geographical boundaries Personalized e-learning has come out to be beneficial in that it presents the content and the learning materials as per the learner's preference and interests. For instance, the same content can be provided in form of a video, or audio based on the learners' learning style where one learner can (Srivastava & Haider, 2020)n use a video lecture and another can use the audio lessons. Intelligent agents have been used to help the learner in their study process by monitoring what the learner prefers in acquiring information (Srivastava & Haider, 2020).

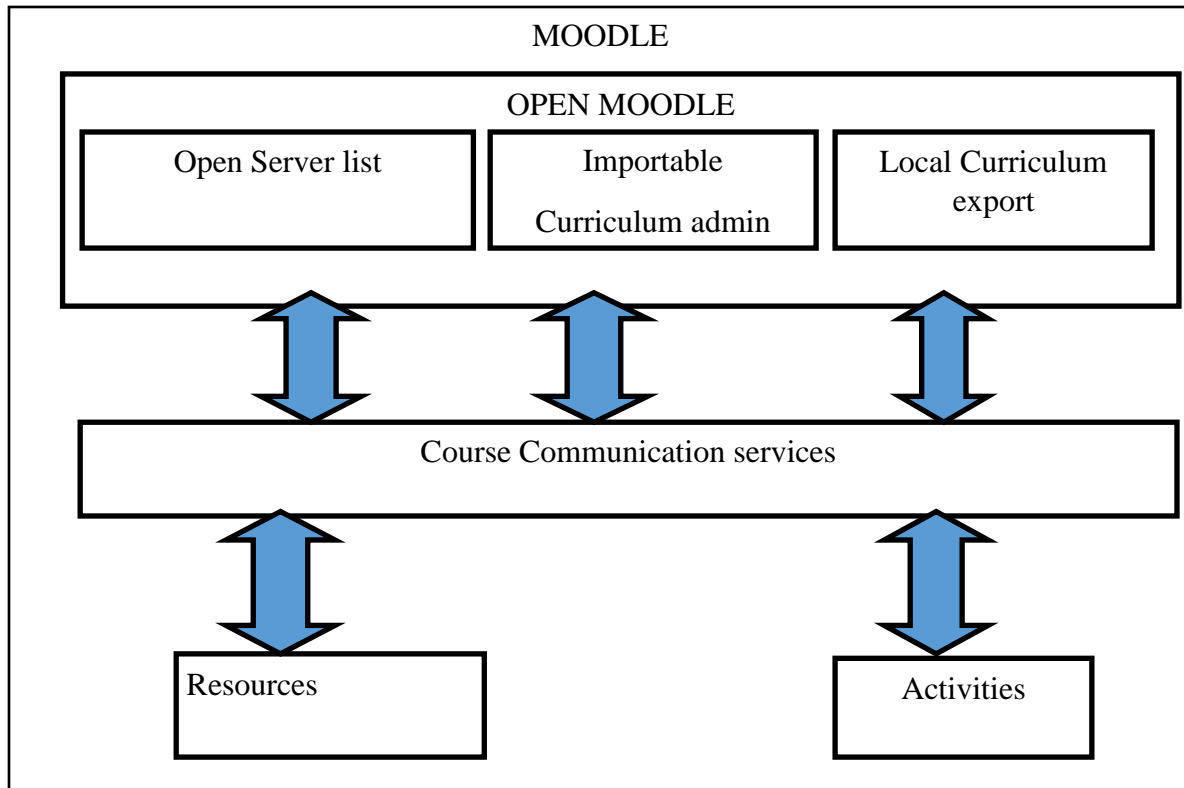
Personalized learning involves organizing the learning resources in a manner that is suitable to the learner's needs. Current research on tailored e-learning focuses on the learner's preferences without taking into account the degree of difficulty and the relationships between various course ideas. Generally, personalization involves instructions that are paced to the learning needs meant for learning preferences and also customized to the target interests of numerous learners. Agbonifo and Olanrewaju (2018) developed a personalized e-learning system based on intelligent agents to meet the learning abilities of various learners. The interface agent provided a friendlier interactive medium for communication and interaction with the other learners. With the help of a genetic algorithm and intelligent agents, it generated an optimal path the learners would consider in the various levels defined by the course content. This demonstrated the importance of course content and curriculum sequencing in personalized e-learning systems (Agbonifo & Olanrewaju, 2018).

Intelligent agents have been used in adaptive e-learning as well as personalized e-learning. Several adaptive e-learning frameworks rely on multi-agent ideas. By tailoring the education process to the demands of the learners, intelligent agent traits like autonomy, proactivity, and cooperation enhance the standard of the educational process. The incorporation of intelligent agents in intelligent tutoring systems is crucial because it puts the learning environment close to the students and emphasizes the study habits of people more than any other technique of instruction (Alexandru et al., 2015). An adaptive e-learning system consists of elements that produce a teaching and learning process that caters to the goals and demands of the learners. It is difficult to adapt e-learning systems to the needs of learners because learning is made simple when a student is actively seeking a solution to a real problem, facilitating the acquisition of extensive knowledge as well as the application of the currently new information to solve issues and the usage of daily activities. As a result, the learning process must be improved to include adaptability and comprehension of the learner's needs (Alexandru et al., 2016). To provide for adaptation in the e-learning system, intelligent agents can be utilized to comprehend and obtain the demands of the learners, such as their expertise, learning preferences, learning ability, and studying motivation and behavior.

## **2.9 Overview of Moodle Learning Management System**

Moodle is an online e-learning management system that is embraced as all-inclusive by various colleges, universities, and some organizations to enhance online learning. Online learning management systems offer elementary tools, which assist in sharing course materials to aid in communication with learners, college assignments, and management grades. Moodle offers additional cooperative and participative teaching and learning environment compared to other learning management systems. It is intended to assist instructors with creating on-the-web

courseware with open doors for rich cooperation. It is open-source and this gives context experts the opportunity to develop additional functionality. E-learning gives the chance learners to communicate electronically with one another just as with their tutors. The interaction between the learners and the tutors can be achieved through email or discussion in chat rooms. (Saraswat, 2014). Figure 2.10 shows the structure of the learning management system ( Moodle)



Source (Saraswat, 2014)

A Learning management system provides interactive learning whereby the learning materials are accessible online, and students receive an automated response for the learners' learning exercises. Present-day trends and best practices in education require the utilization of courseware tools to help the tutoring and learning processes and a productive appraisal. Figure 2.11 below shows a learning management system

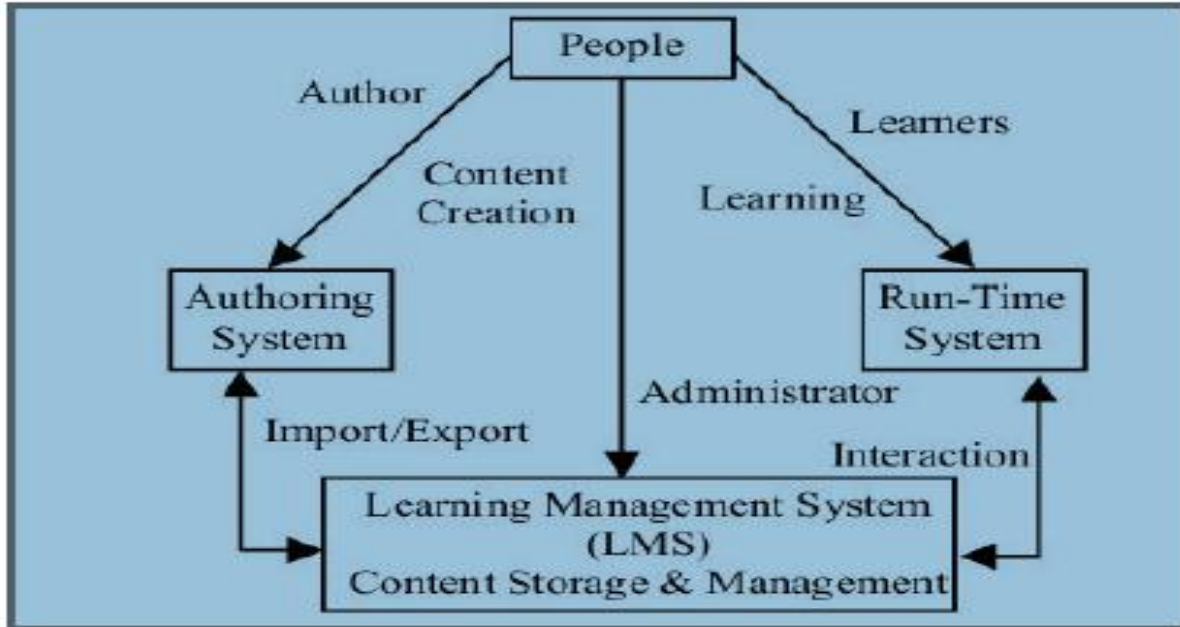


Figure 2.11: Learning Management System Components

Source (Saraswat, 2014)

Moodle is a robust, secure, and integrated online educational platform that has been created to offer teachers, administrators, and students a way to develop learning environments (Kabassi et al., 2015). Moodle provides resources and communication resources. For Moodle to work, it ought to be placed in the main server where necessary configurations are done by the associate administrator to alter access through the internet. It is developed in PHP and SQL databases whereby updates are added from time to time thus it regularly changes and increases. It is template-based to allow the addition of content making its interface interactive and also allowing easy navigation. Moodle has a tool page that is given in a format that is arranged in blocks and sections in which the section has its tool like lessons, quizzes assignments, and forums (Saraswat, 2014).

## **2.10 Theoretical framework**

Figure 2.12 shows a theoretical framework model which consists of three agents; learner agent, system agent, and teacher agent. Learners have to register and have access to the course information then the agents assist them in their learning process. The learner agent helps in analyzing the students' browsing profiles to discover more about their learning status and provides the information to tutors or learners to assist in improving the learners' performance with proper guidance. The teacher agent is used in helping operations in proper timing whenever a learner requires help while the system agent offers their demands to the other agents before collecting the learner information (Duo & Ying, 2012).

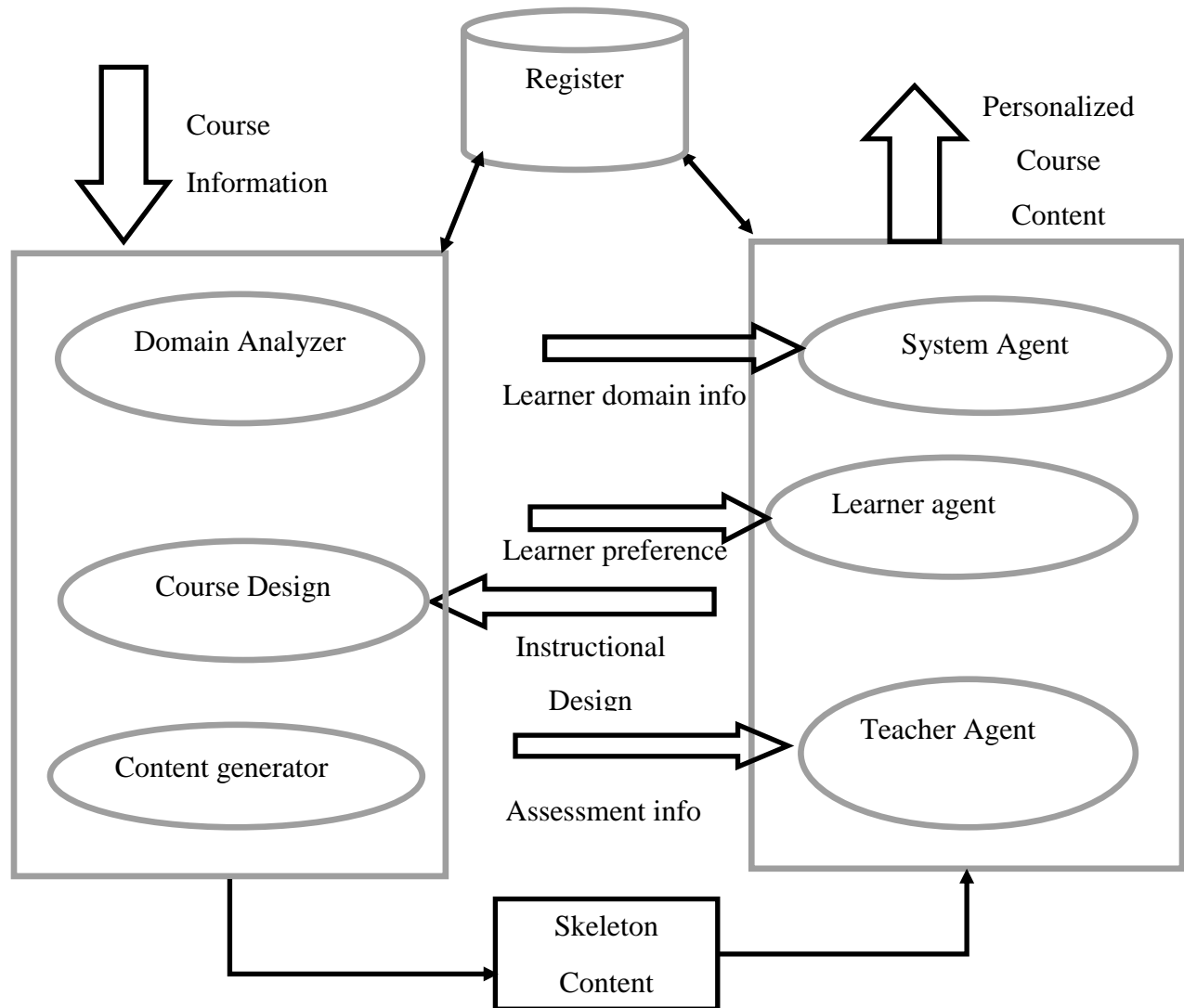


Figure 2.5: Theoretical Framework Model

Source (Duo & Ying, 2012).

Our conceptual model has been developed to cater to different agents as described in the theoretical model. Once these agents are developed they are integrated into a learning management system ( Moodle) as illustrated in figure 2.13

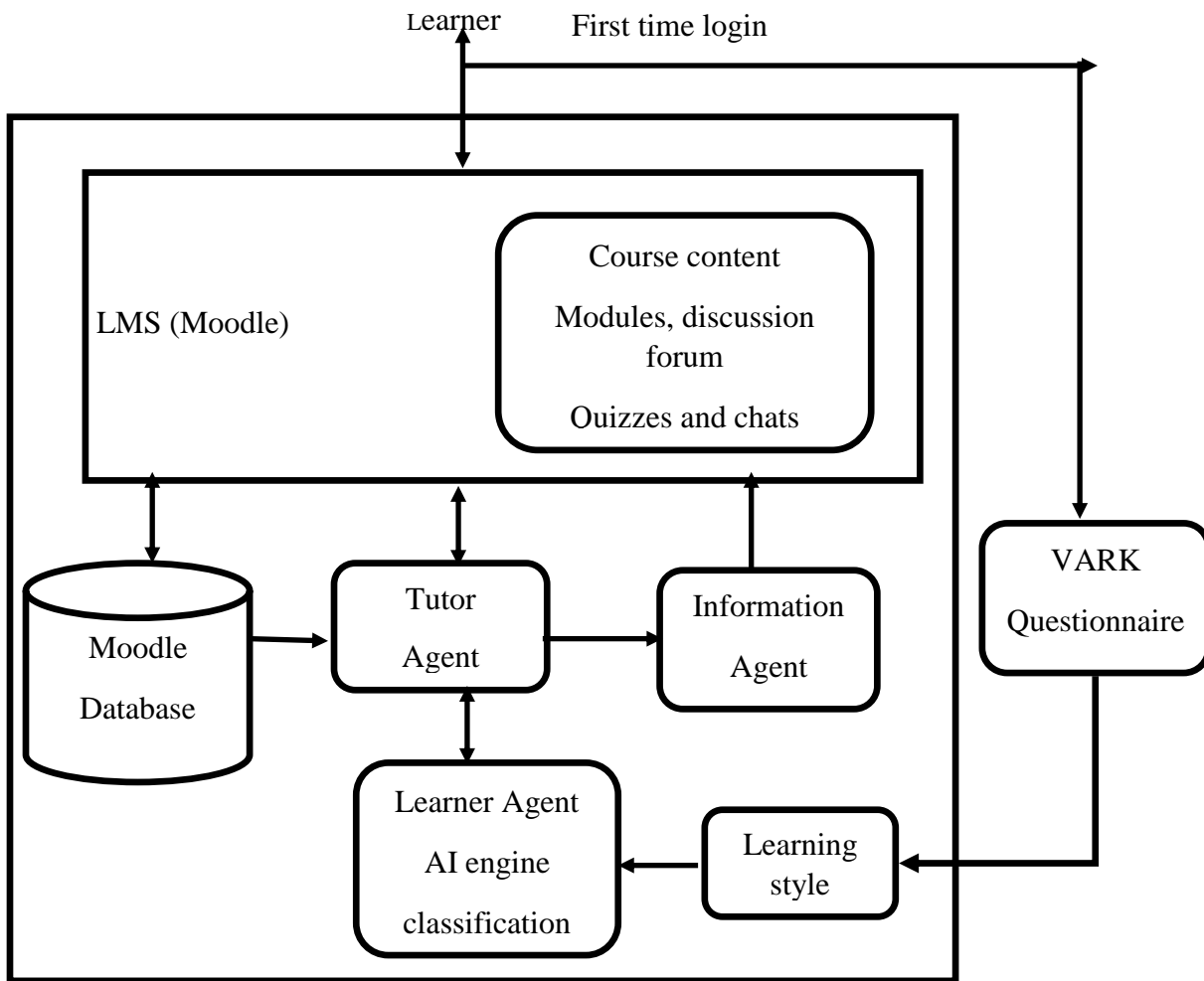


Figure 2.6: Conceptual Framework Model for the intelligent agent-based system

Source (Author)

# CHAPTER THREE

## RESEARCH METHODOLOGY

### 3.0 Introduction

This chapter discusses the methodology that was adopted in the design, development, and implementation of intelligent agents. The following sections are discussed including the research methodology adopted in developing the intelligent agents, the implementation, and how the testing of the agents was done.

### 3.1 Research Methodology

The study adopted Prometheus methodology due to the nature of the system as it supports the development of intelligent agents which uses goals and events as well as the rigor it employs in concentrated design. The Prometheus methodology consists of three major phases as shown in figure 3.1 (Lhafiane et al., 2015).

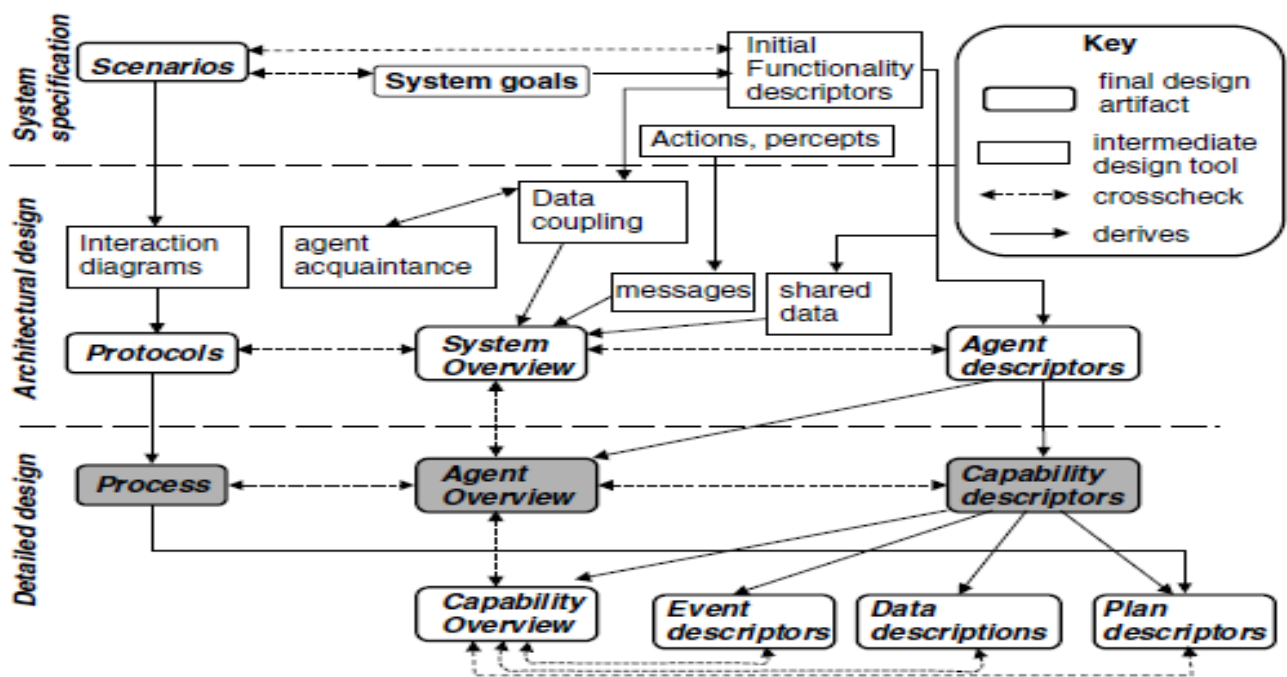


Figure 3.1: Prometheus Methodology Phases

Source (Lhafiane et al., 2015)

### **3.2 System specification/Problem definition**

This phase involved capturing the learning style of the student in terms of how they take in information. They need varied learning approaches because they have various preferences and personal characteristics. The learning style of the student is determined once they login into the Learning management system for the first time. A VARK questionnaire is used which contained 16 questions that the students respond after which they get their personality scores. Once the students fill out the questions the four learning styles have different personality scores and the one with the highest personality score is the one picked as the learning style of the learner using the nested if-else statements.

### **3.3 Architectural Design**

For the architectural design, the event-driven architecture for the agents was adopted. Events are used in this architecture as triggers and for communication between decoupled services and in common applications with microservice which handles specific action. The agents developed are triggered by events depending on any action performed by the learner. For instance, if a learner accesses certain content in the LMS the tutor agent will be triggered by that event and take the necessary action. Figure 3.2 shows interaction diagrams that illustrate a partial picture of the behavior of the system.

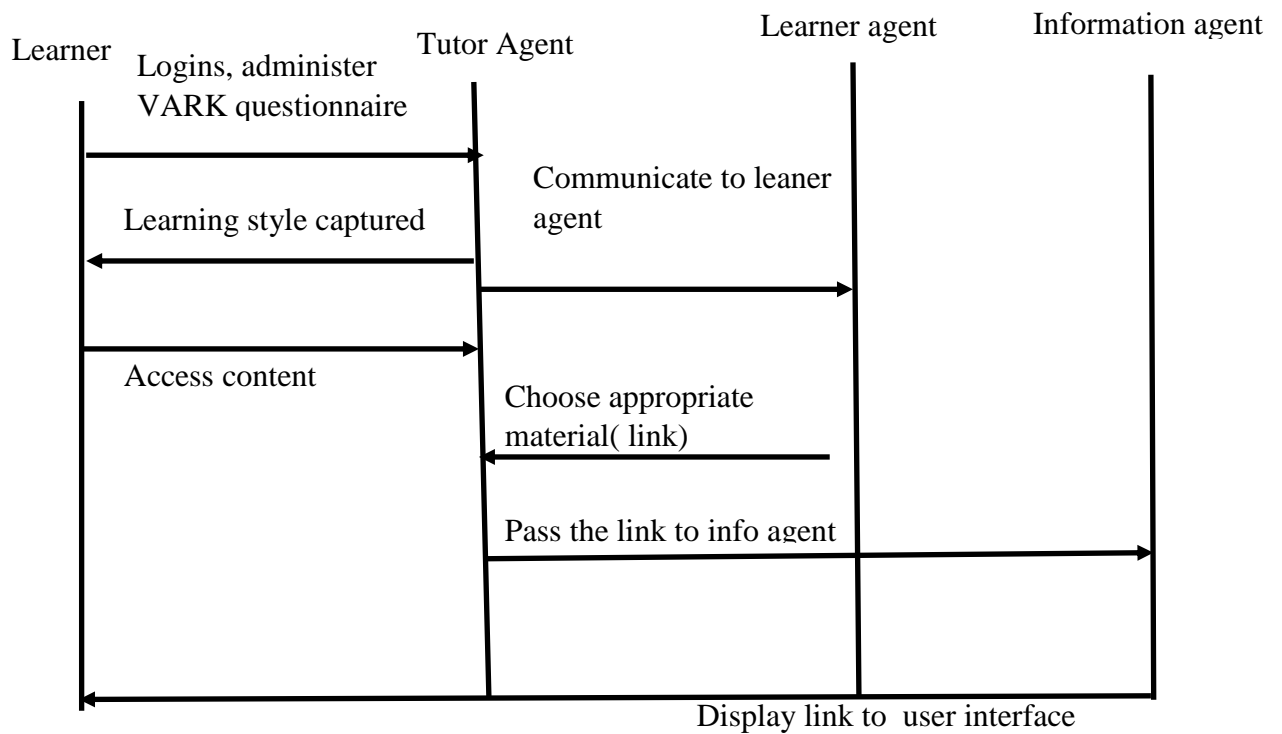


Figure 3.2: Interaction diagram showing the behavior of the system

(Source author)

### 3.4 Detailed Design

This phase aimed at figuring out the manner in which the three agents communicate and interact with one another to get the desired result as well as their internal structure and how each accomplishes its task. The capabilities of each intelligent agent (components of the agent), the internal events as well as their data structures are defined. Each capability is described using a capability descriptor containing information about the external interface to the capability and the event which are produced. The three agents work together with constant communication with each other. The tutor agent is the central agent, which coordinates the learning. The learner agent, which has the AI engine, recommends additional learning materials for the learning style.

The information agent displays the recommended materials to the student in form of a link.

Figure 3.3 shows how the three agents interact with one another.

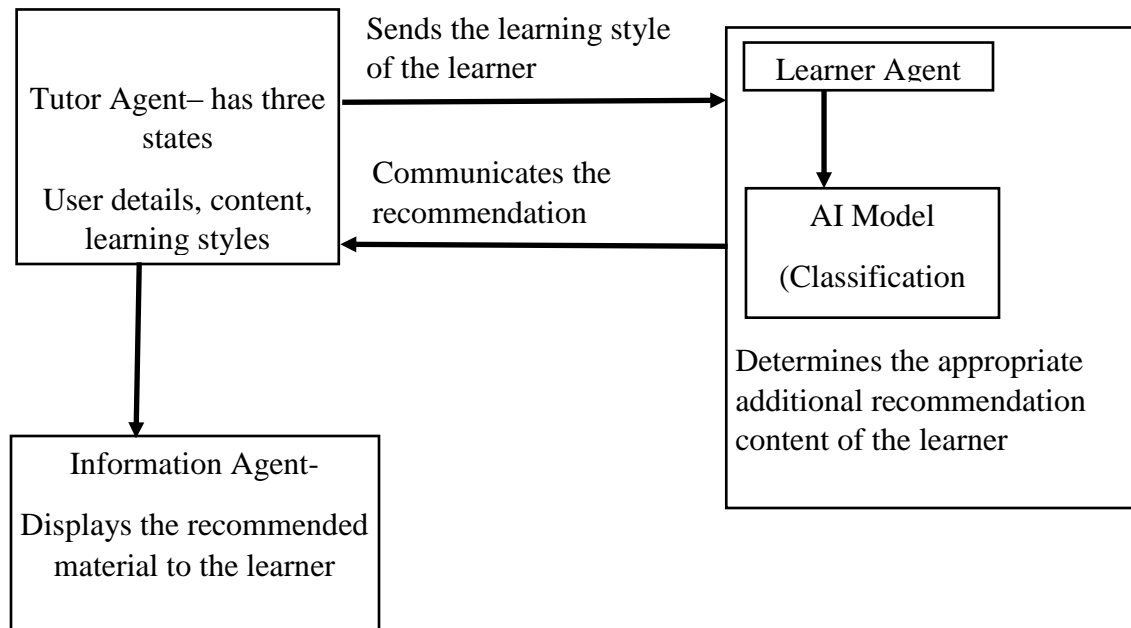


Figure 3.3: The three agents and how they interact with one another

Source ( Author)

### 3.5 Implementation

Python programming was used to develop the learner agent whereby TensorFlow framework was used. Various libraries were also used including tflearn which is a library in TensorFlow that provides a higher-level API to TensorFlow to facilitate and show new experiments. PHP programming language was used to develop the tutor and the information agent. The reason for using PHP was for compatibility reasons as Moodle is developed using PHP to avoid compatibility issues when integrating them into the Learning management system. Other tools included Google Colab which provided the environment to write the code and execute them and a learning management system (Moodle) to integrate the agents.

### **3.6 Testing**

For testing purposes, the VARK questionnaire is incorporated into the learning management system to capture the learning style of the student. The questionnaire contains 16 questions that the students answer once they log into the system for the first time. To test the intelligent agents they are first integrated into the learning management system. The agents are integrated into the learning management system as a plugin. Then a programming course (C programming course) with four topics namely: arrays, data types, control structures, and functions are created to initiate the learning process and check whether the agents are working as expected.

## CHAPTER FOUR

### RESEARCH FINDINGS AND DISCUSSIONS

#### 4.0 Introduction

This chapter describes how the learning style is identified, how the agents are designed and implemented, and finally how they are integrated into Moodle. The system is a multi-agent system that consists of three agents that communicate with each other to achieve a certain goal i.e. assisting the learner in their study and providing a recommendation to the learner based on their learning style. The agents collaborate to provide a personalized learning experience. The chapter is organized as follows: determination of the learning style of a learner, description of the design and development of the three agents, system architecture, and system implementation.

#### 4.1 Determining the Learning Style of the Learner

Learners' experiences are enhanced by the use of learning styles in e-learning environments because the knowledge is retained in a more realistic and efficient approach. Because learning style is a learner attribute, the learner can acquire skills, knowledge, and attitudes by studying or experiencing according to their learning style choice in adaptive e-learning environments. There are numerous learning styles presented by various educational pedagogies which assist in capturing the learning style of any given learner. The study adopted the VARK learning style model due to various reasons. . It is acceptable by experts and suitable for the majority of educational systems. Additionally, it is quite straightforward to use and grasp the outcomes (Daoruang et al., 2018). Also, it largely depends on a person's personality type (Kumar et al., 2017) which is appropriate for us because we're interested in the learning traits ( learning style) of the students. In addition, VARK does not, in any case, impact the structure and the sequencing

or learning materials resented in the e-learning platform, Furthermore, it is intrusive and has the capability to clearly and effectively identify, collate and map the requested learning material type. Due to the reason presented, VARK qualifies as the ideal tool for this study. The VARK questionnaire is integrated into the LMS (Moodle) as an application module. The module analyzes the responses of the learner and establishes the learning style of the student. The learner at the initial interaction with the Learning management system is taken through the learning style module and answered the sixteen questions in the questionnaire as given by Flemming. The learner agent captures the learning style of the student and is stored in Moodle database. The learning style of the student is determined by checking the learning style with the highest points from answering the sixteen questions. The personality score is reached by using nested if statements to identify the learning style having the highest rank. The following pseudocode has been implemented to help identify the learner's preferred learning style:

### **Pseudocode for Determining the Learning Style**

1. Student answers the 16 questions in the questionnaire
2. if max\_points = r, then your learning style is to read and write
3. if max\_points = v, then your learning style is visual
4. if max\_points = a, then your learning style is auditory
5. if max\_points = k, then your learning style is kinesthetic

The following shows a code section of the above pseudocode

```

$max = max($v,$a,$r,$k);
  if($max == $v)
  {
    $profile = 'visual';
    $points = $v;
    $desc = $vis;
  }
  else if($max == $a)
  {
    $profile = 'auditory';
    $points = $a;
    $desc = $aud;
  }
  else if($max == $r)
  {
    $profile = 'read/write';
    $points = $r;
    $desc = $rw;
  }
  else if($max == $k)
  {
    $profile = 'kinesthetic';
    $points = $k;
    $desc = $kin;
  }

```

## 4.2 Design and Development of the Intelligent Agents

To accomplish their intended goals, agents—encapsulated computer systems that are located in specific contexts—take flexible, autonomous action in those surroundings. Numerous researchers define agents as elements that work together to construct multi-agent systems. To achieve their individual or collective goals, many agents engage in cooperative or competitive interactions with one another and their environs. A multi-agent system is an interactive one that comprises several specialized agents that may both create their stimuli and respond to those produced by other agents. It's crucial to specify the agents' internal behavior, knowledge, and interactions with other agents in a multi-agent system when there are multiple agents involved (

Russell & Norvig, 2020). This study involves three agents that interact and communicates with each other to assist the learners in their learning process as shown in table 4.1.

Table 4. 1: Summary of the various intelligent agents

<b>Agent</b>	<b>Function</b>
Learner Agent	Have the AI(Artificial Intelligence) engine that uses a deep neural network which helps in the recommendation of the appropriate learning resources to the learners according to their learning style
Tutor Agent	Coordinates learning in the system. It communicates and interacts with the learner and the information agent. It has access to the Moodle database whereby it has the details of the student( student profile) Tracks the learning process of the student/ activities the student performs in the system and communicates that to the learner agent for a recommendation
Information Agent	Displays the recommended materials to the Moodle user interface which it receives from the tutor agent.

#### 4.2.1 The Learner Agent

This agent has an AI engine that makes recommendations for the best learning resources for students based on their preferred learning style using a deep neural network. This agent recommends the appropriate learning material based on the learner's learning style. Figure 4.1 shows a flowchart to illustrate the procedure used that was used to come up with the learner agent.

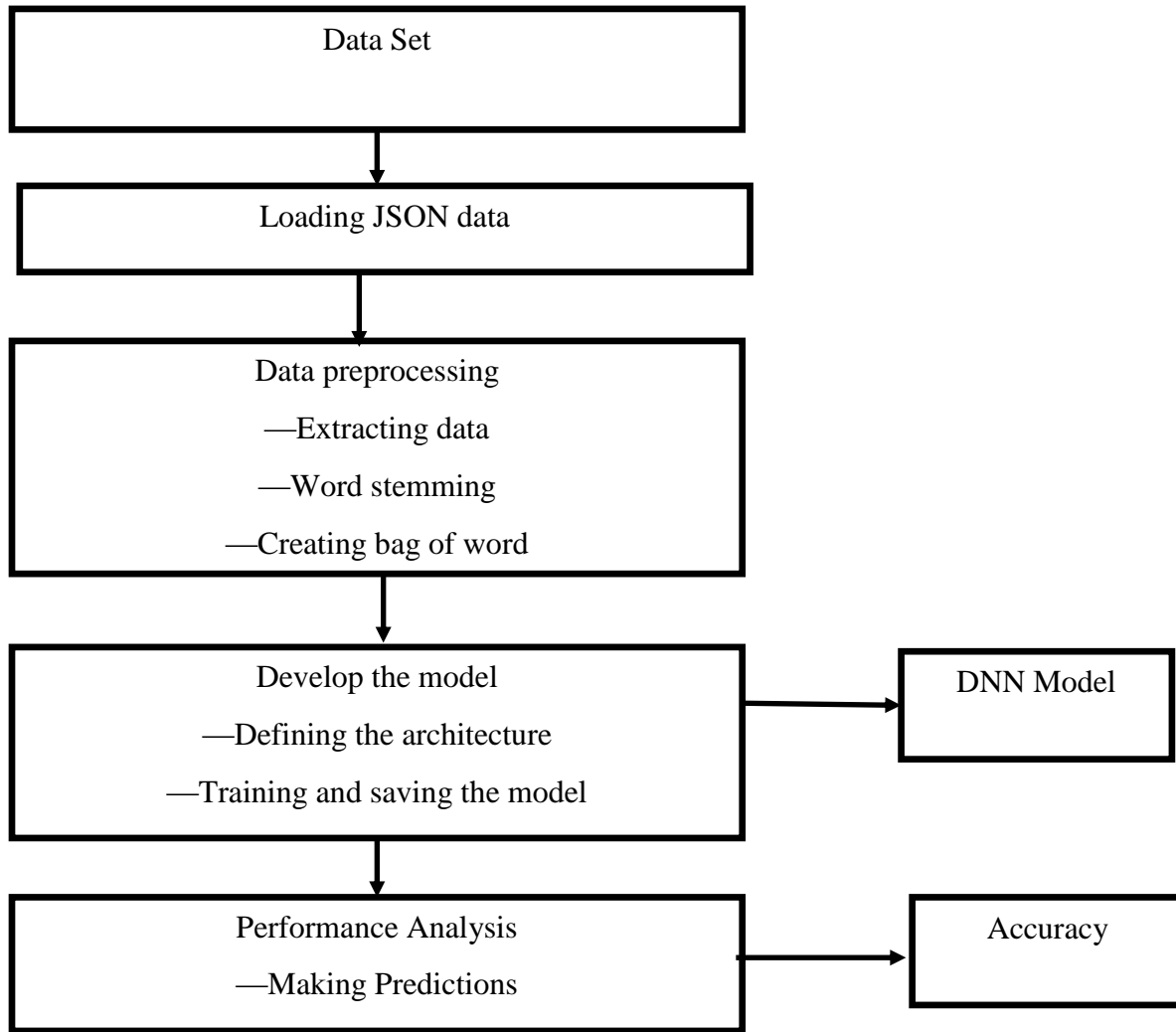


Figure 4. 1: Machine learning process used to create the learner agent

(Source author)

### Training Data Set

Simple data was collected from the web and converted into a JSON file that was used to create a group of messages the user (learner) is expected to access, mapped to a bunch of suitable responses. Each dictionary in the file has a tag (topic) that identifies the category to which every subtopic belongs. The JSON data was utilized to train a neural network that analyzes sentences of text and categorizes them according to one of the topics (tags) in the file. The network then

gathers feedback from the groups and displays it to the user. The data contained three parameters/features including topic, subtopic, and responses. From the sample dataset shown in figure 4.2, the tag represented topics while the pattern represented the subtopic. Figure 4.2 shows sample data of the data used for training.

```
{
  "intents": [
    {
      "tag": "arrays read_write",
      "patterns": ["What is an array read_write?", "Array index read_write", "array Declaration read_write", "Assigning or storage read_write of values in array", "Accessing array elements read_write"],
      "responses": ["Array elements are accessed using an index value: see <a href='https://www.geeksforgeeks.org/arrays-in-c-cpp/'>this link</a>", "An array is a data structure that consists elements. See <a href='https://techterms.com/definition/array'>this link</a>", "The declaration and stotage of elements in an array is as shown. See <a href='https://www.tutorialspoint.com/cprogramming/c_arrays.htm'>this link</a>", "An array is a group (or collection) of same data types. See <a href='https://beginnersbook.com/2014/01/c-arrays-example/'>this link</a>", "there are diferent ways of declaring arrays. See this article in <a href='http://ee.hawaii.edu/~tep/EE160/Book/PDF/Chapter7.pdf'>this article</a>", "to read more on array index see the following slide . <a href='https://www.slideshare.net/kaushal_kush/array-ppt'>this link</a>"],
      "context_set": ""
    }
  ],
}
```

Figure 4.2: Sample data from the JSON file

Source (Author)

## Data Extraction

After the JSON data is loaded data preprocessing was done. This involved taking out the data that was needed in the JSON file i.e the subtopic and the corresponding topics to which they belonged and a list of unique words in the subtopic and then setting up a blank list to store them as shown below

```
words = []
labels = []
docs_x = []
docs_y = []
```

This was achieved by looping through the JSON data. Instead of having them as strings, using the nltk.word tokenizer, each pattern was transformed into a list of words. Each pattern was added to the docs\_x and its corresponding tag into the docs\_y list as shown in the code section below

```
for intent in data['intents']:
    for pattern in intent['patterns']:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)
        docs_x.append(wrds)
        docs_y.append(intent["tag"])

    if intent['tag'] not in labels:
        labels.append(intent['tag'])
```

Figure 4.3 below shows a simple tokenization

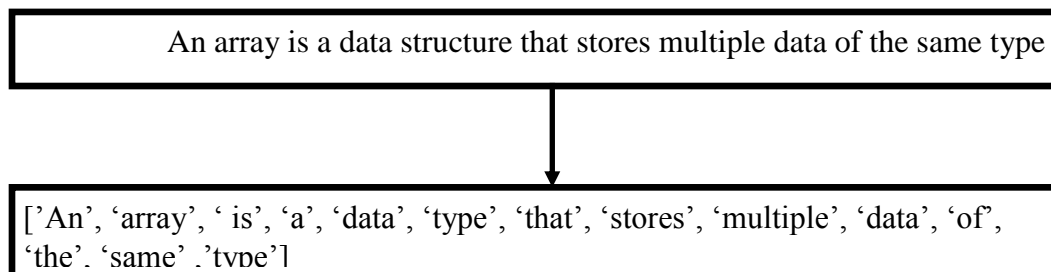


Figure 4. 3: A simple tokenization

Source (Author)

The next step of the data preprocessing was word stemming

## Word Stemming

In order to stem a word, one must look for the word's root. For instance, the word "what's" stem may be "what," and "learning's" stem could be "learn". This process was used to reduce the vocabulary of the model and attempt to find more general meaning behind the sentences as shown in the code section below

```
words = [stemmer.stem(w.lower()) for w in words if w != "?"]
words = sorted(list(set(words)))
labels = sorted(labels
```

The stemming process aimed to create a unique list of stemmed words for further processing. The following shows a sample of stemmed words

```
['access', 'an', 'and', 'argu', 'array', 'assign', 'audit', 'bas', 'c', 'cal',
```

The next step of the data preprocessing was creating a bag of words.

### **Bag of Words**

Since neural networks and machine algorithms require numerical input a bag of words was created to represent the sentences(text) in numbers. Creating a bag of words involves representing each text with a list of the length of the number of words in the vocabulary. This process consists of two steps: Namely:

- i) Defining the vocabulary- set of all words in the sentence/text
- ii) Vectorization-counting how many times each word appears

For instance let's consider the following example with three sentences S1, S2, and S3

S1-What is an array, S2-An array is a data structure and S3-An array consists of an element

The vocabulary from the three sentences are: {*what, an array, is, a, data, structure, consists, of, element*}. The result of vectorizing is as shown in the



```

[[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]]

```

Each position in the list represented one unique tag i.e a 1 in any of those positions indicates the tag (topic) represented. This is illustrated in the following code section

```

training = []
output = []
out_empty = [0 for _ in range(len(labels))]
for x, doc in enumerate(docs_x):
    bag = []
    wrds = [stemmer.stem(w.lower()) for w in doc]
    for w in words:
        if w in wrds:
            bag.append(1)
        else:
            bag.append(0)
    output_row = out_empty[:]
    output_row[labels.index(docs_y[x])] = 1
    training.append(bag)
    output.append(output_row)

```

After creating the bag of words the training data and the output are then converted to NumPy arrays as illustrated below

```

training = numpy.array(training)
output = numpy.array(output)

```

## Developing the Model

After the preprocessing, the next step involved training the model with the data created. A fairly standard feed-forward neural network with two hidden layers was used. A feed-forward network receives an input to produce some kind of output to make a prediction. The goal of the network was to look at the bag of words and get the class/tag/topic that they belong to (one of the tags(topic) from the JSON data. The following code section defines the architecture of the model.

```
net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)
model = tflearn.DNN(net)
```

The model consists of four layers, the input layer, the output layer, and two hidden layers. The hidden layers are fully connected with 8 neurons as shown in figure 4.4

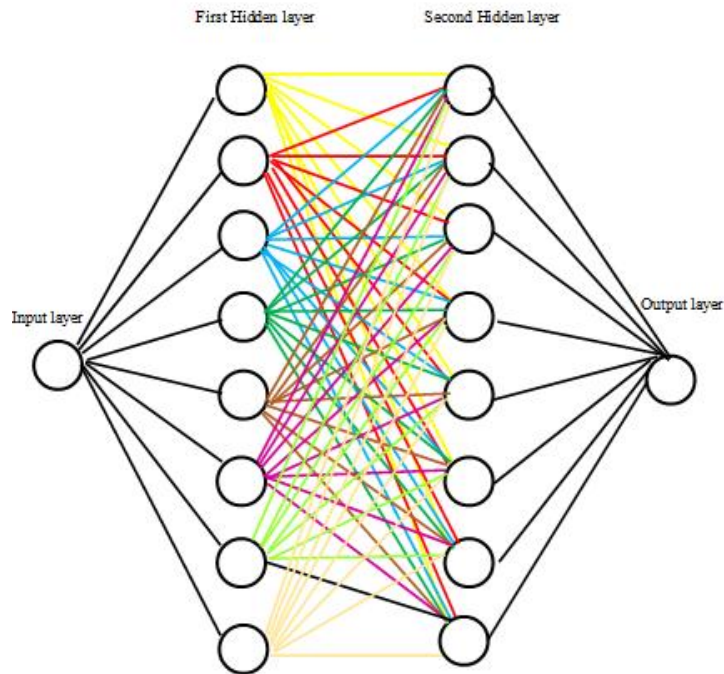


Figure 4. 4: A neural network consisting of 8 fully connected neurons and two hidden layers

Source (Author)

The input layer consists of the input data which is the length of the training data. The output layer contains the neurons representing each of the classes/topics. It also contains the softmax activation function in which the neurons in the output layer are all run through this activation to give a probability of each neuron. A response is picked from the neuron with the highest probability. Generally what the model/agent is doing is predicting which tag/topic to take a response from and to give to the learner/user. The process involved is taking in a bag of words as input and getting labels/topics as output indicating where the response should come from. For instance, if the sees the text array it starts to change the weights and biases so that “what is an array” comes from the responses in the topic arrays. The whole idea is classifying the sentence of words (text) to some kind of a topic that the learner may be accessing. After setting up

the model the next step was to train and save the model by fitting the data into the model as illustrated in the code below

```
model.fit(training, output, n_epoch=1000, batch_size=8, show_metric=True)  
model.save("model.tlearn")
```

Other parameters/metrics used in the model included the number of epochs, the batch size, the loss, and the optimization algorithms. The number of epochs set indicates how many times the model will be exposed to the same data while being trained (the number of iterations). One epoch signifies that each sample in the training dataset has had its internal model parameters updated. The batch size is the number of samples that are sent across the network in a single batch. The loss evaluates how well the neural network models the training data by comparing the target and predicted output values. The aim is to minimize the loss between the predicted and the target output. The optimization algorithm accelerates the gradient descent algorithm into consideration the exponentially weighted average. The averages make the algorithm converges towards the minima very fast.

The following illustrates the mentioned parameters: epochs=1000, Batch size= 8, loss=0.0749, accuracy =0.9970 and optimization algorithm = Adam as shown below figure 4.5

```
Training Step: 11000 | total loss: 0.07490 | time: 0.078s  
| Adam | epoch: 1000 | loss: 0.07490 - acc: 0.9979 -- iter: 84/84
```

Figure 4. 5: Results of the neural network metrics used

Source (Author)

## Making Predictions

Finally, the last step was making predictions whereby ideally it involved generating a response to any text (sentence of words) a user types in (what the learner is accessing from the course content). Recall that the model produces a list of probabilities for each class rather than merely spitting out sentences. Thus the process of generating a response is as follows:

- i) Getting some input from the user(in our case checking what the user is accessing in the course content)
- ii) Converting it into the bag of words
- iii) Getting the prediction from the model
- iv) Finding the most probable class/tag
- v) Then picking a response from that class(topic)

A `bag_of_word ()` function was created to transform the string input to a bag of words using our created word of the list as illustrated in the code below

```
def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]
    s_words = nltk.word_tokenize(s)
    s_words = [stemmer.stem(word.lower()) for word in s_words]
    for se in s_words:
        for i, w in enumerate(words):
            if w == se:
                bag[i] = 1
    return numpy.array(bag)
```

And then a predict () function is also created to get a prediction from the model and grab an appropriate response from the JSON file of responses as illustrated in the following code

```
results = model.predict([bag_of_words(inp, words)])
results_index = numpy.argmax(results)
tag = labels[results_index]
for tg in data["intents"]:
    if tg['tag'] == tag:
        responses = tg['responses']
```

### **How the Learner Agent works**

Begin (Learning process begins)

1. Once the student logs into the system
2. Keep waiting/checking for any communication from the tutor agent regarding the learning style of the student
3. Once it gets the learning style from the tutor agent, use the AI engine to predict the appropriate recommendation material (response) for the learner based on the learning style
4. Passes the recommendation to the tutor agent which then exchanges messages with the information agent
5. Repeat steps 2 to 4 for any other student who is logged into the system

End

### **4.2.2 Tutor Agent**

The tutor agent coordinates the learning in the intelligent system. It communicates with the other agents in assisting the learner in their learning process. This agent has access to the Moodle database and therefore is aware of the profile of the students. This means it is aware of the three

states of the student including the details of the user (user id, the learning style, and the content accessed). It then passes these details to the learner agent for the recommendation of the appropriate content based on their learning style. The following shows the pseudocode of the tutor agent with the code provided in appendix II.

### **How the Tutor Agent Works**

Begin (Learning process begins)

1. Once the student login
2. Check the state of the student and the learning style of the student
3. Identify what the student has accessed
4. Pass that information to the learner agent
5. Wait for a response for the learner agent
6. Repeat steps 2 to 5 for every student who logins
7. Upon receiving the feedback from the learner agent pass the information to the information agent

End

### **4.2.3 Information Agent**

The work of the information agent is to display (notify) the recommended materials to the student on the user interface. The tutor agent communicates also with this agent on the particular resources to recommend to the learner. This agent also is aware of the state of the students in terms of which page is the student currently viewing and where to exactly display the recommended material. The following shows the pseudocode of the information agent with the source code for the agent also provided in Appendix II.

## **How the Information Agent works**

Begin (Learning process begins)

1. Once the student logs into the system
2. Keep waiting/checking for any communication from the tutor agent
3. Once it gets a notification from the tutor agent
4. Display the feedback of the student on the user interface
5. Repeat steps 2 to 5 for any other student who is logged into the system

End

## **4.3 System Architecture**

The first step involves the identification of the learner's style through the VARK model. After logging in, the students fill out the VARK questionnaire to determine each student's preferred learning style. The student should be aware of their responses and how they will be used. This is to guarantee that the questionnaire will receive an accurate response. The system uses the responses to select the resources that best suit their learning preferences. Based on the learners' learning preferences, learning resources are recommended to them. By offering learning resources that are relevant to the goals and learning styles of the students, personalized e-learning is supported. The learner agent explores the JSON file for the content based on the learners' learning accomplishments and learning styles. For example, if the learner's learning style is read/write, the system suggests content via a link in the form of a pdf, presentation slides, textbook, or lecture notes.

The three agents are incorporated into the Moodle backend, where the database is accessed to retrieve learner data, according to the design. The Moodle database receives an additional table

to contain the student's recorded learning style. The table below shows a description of the additional table in the Moodle database

FIELD	TYPE	NULL
id (Primark key)	int (11)	NO
userid	int (11)	YES
profile	varchar(20)	YES
points	int (11)	YES
description	Text	YES

The tutor agent interacts and communicates with the other agents, especially the learner agent and the information agent, to coordinate learning within the system. It has information about the students (student profiles), including their names, user IDs, and preferred methods of learning, because it has access to the Moodle database. After the student begins their studies, the tutor agent checks what their access is to elements like the uploaded modules, assignments, and discussion forums. The learner agent is then informed of this information. The deep neural network, incorporated into the learner agent predicts the most appropriate content in accordance with the student's preferred learning style. The tutor agent then relays the predicted learning material to the information agent. The learner gets feedback from the information agent via the Moodle user interface. Figure 4.5 shows the system architecture of the intelligent system.

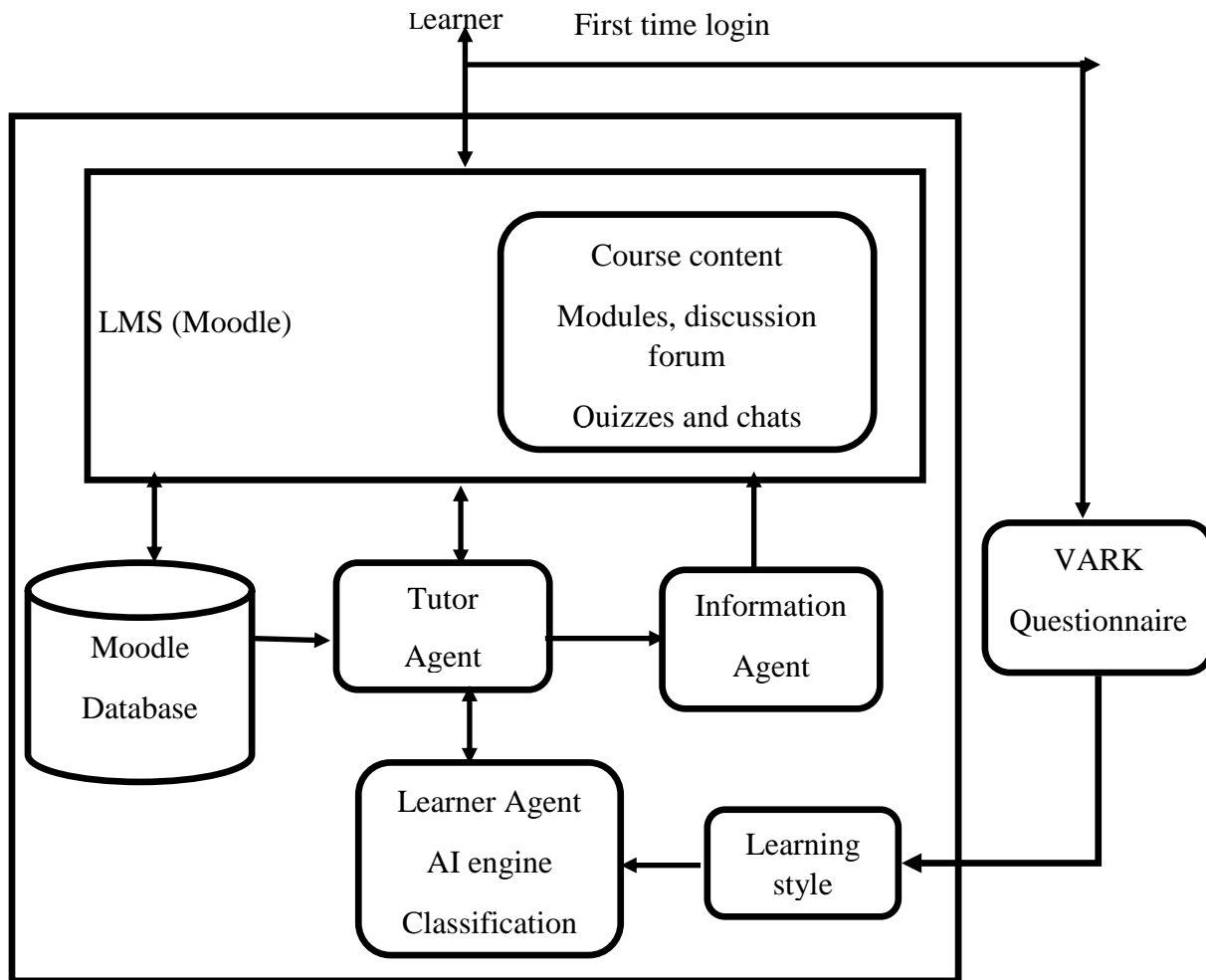


Figure 4.6: System architecture

Source ( Author)

#### 4.4 System Integration

The three agents were integrated into a learning management system (Moodle) as a plugin. The VARK questionnaire was also integrated into Moodle for capturing the students' learning styles the first time they log into the system. The tutor agent plays the role of the central agent by coordinating the learning process. It monitors the actions of the learner by checking what the student access in the course content. The learner agent liaises with the tutor agent to know the learning style of the student to provide the necessary additional learning materials for the learner.

The information agent presents the suggested content to the learner through the LMS user interface.

#### **4.5 Results from the Learning management system**

##### **i) Learning Style Identification**

For testing purposes, a course (Introduction to programming in C) with four topics—arrays, data types, functions, and control structures in C—is created in Moodle. The VARK questionnaire is the first element a learner completes after enrolling in the course for the first time. As seen in figure 4.6, the questionnaire is only given once when a learner logs in for the first time.

student one ▾

# Test Environment

---

Home Dashboard Events My Courses
Hide blocks

Your learning style will be identified shortly. Redirecting...

Your learning style could not be determined. You have been redirected to the questionnaire. Please fill in the questionnaire below to the best of your ability. This is only a one time thing.

## VARK Questionnaire

Please choose the most suitable choices that describe best the asked question

1. I need to find the way to a shop that a friend has recommended. I would:

- a.  find out where the shop is in relation to somewhere I know
- b.  ask my friend to tell me the directions
- c.  write down the street directions I need to remember
- d.  use a map

---

Home Dashboard Events My Courses

- b.  from somebody who talks it through with me
- c.  using a written description of my results
- d.  using examples from what I have done

15. I want to find out about a house or an apartment. Before visiting it I would want:

- a.  to view a video of the property
- b.  a discussion with the owner
- c.  a printed description of the rooms and features
- d.  a plan showing the rooms and a map of the area

16. I want to assemble a wooden table that came in parts (kitset). I would learn best from:

- a.  diagrams showing each stage of the assembly
- b.  advice from someone who has done it before
- c.  written instructions that came with the parts for the table
- d.  watching a video of a person assembling a similar table

Get Personality Score

Figure 4.6: Sample of the VARK Questionnaire

Source (Author)

**i) Notification of the Learning Style**

They are informed about their learning style after responding to every question on the questionnaire with a brief description of the learning style. For instance, the learning style of one of the students testing the system as indicated in figure 4.7 show their learning style is auditory.

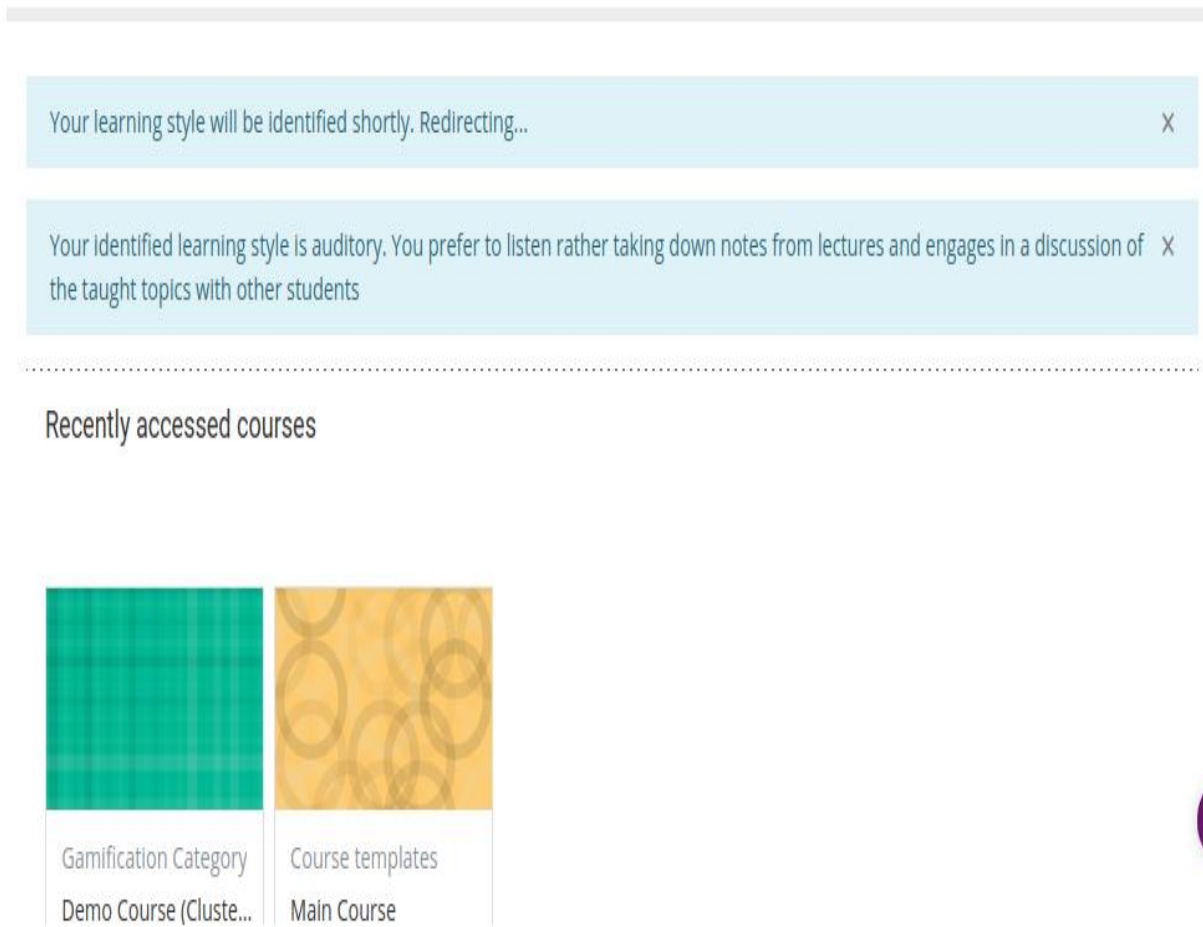


Figure 4.7: Notification of the Learning style

Source ( Author)

## ii) Personalized Learning

They are given recommendations for additional materials based on the course content they access and their preferred learning style. For example, if their preferred method of learning is auditory, they are suggested a video link to watch and listen to, as seen in figure 4.8.

The screenshot shows a user interface for a course titled "INTRODUCTION TO PROG ...". The navigation bar includes links for Home, Dashboard, Events, My courses, and This course. A breadcrumb trail indicates the user is in "Dashboard > My courses > SIT > ARRAYS > What is an array". A light blue notification box contains the text: "Definition of an array . Listen to the following. from this link" with a close button (X). Below this, the page title "What is an array" is displayed. The main content area contains the following text:

Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

Figure 4.8: Sample of the feedback/recommendation

Source (Author)

## 4.6 Discussion of Research Findings

The results of the discussion are presented according to the following research objectives that were tested in the research

- a) Objective 1: Design a learner, tutor, and information agent that will support personalized e-learning
- b) Objective 2: Develop and implement a learner, tutor, and information agent that will support personalized e-learning
- c) Objective 2: Integrate the intelligent agents into a learning management system

### **Objectives 1 and 2: Design, develop and implement a learner, tutor, and information agent that will support personalized e-learning**

Prometheus methodology was adopted to develop the intelligent agents as it provided concentrated design and as well as the nature of the intelligent agents. In this study, the results of the intelligent agents have shown that intelligent agents can be used to support personalized e-learning. Personalized e-learning is based on one's learning style whereby a student can be able to get learning materials based on how they consume information. The findings align with research carried out by Zagulova et al., (2019) who developed a personalized intelligent multi-agent system for engineering courses based on their learning style and other needs using the Filder Silverman model to determine their learning style. From the study, the results indicated that agents assisted the learners in their learning process which involved first identifying the needs of the student in terms of how they consume information. In addition, this study also aligns with a study carried out by Fasihfar and Rokhsati (2017) on the application of intelligent agents in e-learning systems that found that intelligent agents support learners in their learning process by continually providing content that the learners with they opt for. (Melesko & Kurilovas, 2017)

designed and developed a personalized intelligent multi-agent system for engineering courses which was also identifying student learning styles using the Filder Silverman model and found out that recommendation of the student's preferred content/ learning materials can be recommended based on the student learning style which as well aligns with our study findings.

**Objective 3: Integrate the Intelligent agents into a Learning Management System.**

The three agents were successfully integrated into a learning management system as a plugin. Since intelligent agents can't work independently because they require an environment where they can act on, we used Moodle where the agents were installed as a local plugin. Once the agents were integrated the next step was to check whether the agents were working as expected. The results indicated in figure 4.2 and Figure 4.3 indicated the agents were working after being integrated into Moodle.

## **CHAPTER FIVE**

### **CONCLUSION AND RECOMMENDATIONS**

#### **5.0 Introduction**

This chapter describes the conclusion of the research findings based on the research objectives, recommendations, and limitations of the study. The main objective of the study was to design and implement an intelligent system for supporting personalized e-learning.

#### **5.1 Summary of Research Findings**

From the results discussed, the study was successful as all the objectives indicated in chapter one were all achieved. The agents can assist the learner in the learning process. The agents were developed and implemented to determine whether they were performing as expected. Finally, they were integrated into Moodle as a plugin. Personalization was supported by identifying the learning style of learners to first understand how they prefer to consume information in their learning process. The results indicate that it is possible to train a learner agent using deep neural networks and provide personalized learning to the learner based on the learning style.

#### **5.2 Conclusion**

The three agents were successfully designed, developed, implemented, and successfully integrated into a learning management system (Moodle) to provide personalized e-learning based on the learners' learning styles. The results showed that intelligent agents can be used to support personalized e-learning. The design and the development of the three agents was achieved by the use of the Prometheus methodology which is the most common methodology for developing an agent-based system from the literature review done. The intelligent system can be used by learners in higher learning institutions to study as per their preferred method of learning.

Through learning management systems like Moodle, the integration enables students to learn according to their preferred learning style.

### **5.3 Recommendation of the study**

The study recommends the following recommendations. In developing the learner agent which had the AI engine, only a deep neural network model was used for predicting the learning material to recommend to the learner. Other machine learning models/ algorithms are recommended to get the best results instead of relying on only one model/algorithm. An existing tool was used to determine the learning style of the student due to the time factor and thus a different way of determining the learning style is recommended whereby learners are exposed to a learning management system for a whole semester and their data from the learning management system is mined to predict and determine their learning style. Consequently, other learning style models like as Felder- Silverman model, and the Kolb learning style model are also recommended to compare the results. Additionally using a large dataset is also recommended for instance getting a recommendation from various sources on the web. Also, performing an experimental study on how intelligent agents are effective in supporting personalized e-learning by using a control group that uses the intelligent system and comparing it with another group not using the intelligent system.

### **5.4 Limitations of the Study**

The recommendation of the additional learning materials is only limited to the data in the repository (JSON file). In addition, the recommendation is only based on the learning style of the student. Also, one learning style model (VARK) is used to determine the learning style of the learner. Furthermore, instead of limiting the identification of the learning style to only one learning style model, automatic detection of the learning style based on data in a learning

management system could also have produced better results as the learning style of a student is subject to change over time. The automatic detection of the learning style has the advantage in that there is no supplementary amount of time required for instance filling questionnaire, as this approach gathers the information from a span rather than a specific point in time, and therefore changes in the characteristics of the student can be easily followed. Furthermore, it also uses real-time data as the learner interacts with the system therefore it is more accurate and less error-prone. In addition, it also makes the learner focus more on learning instead of making them waste time answering questionnaires and providing feedback.

### **5.5 Further Research**

The following further studies are suggested to address the limitation of the study. Instead of using one characteristic of the student (learning style) other features such as the learning path of the learner, the affective state as well as adaptive learning are suggested to be included in further studies. Also, to address the limitation of using one learning style to establish the learning style, a combination of other learning styles models like the Felder-Silverman, and Kolb learning styles models among others could produce better results. Consequently, rather than using an existing tool like the VARK questionnaire to determine the learning style, automatic detection of the learning style is suggested whereby the learning style is mined from the data collected in a learning management system. A comparison of the results using an existing tool to capture the learning style and automatically detect the learning style using machine learning techniques is suggested for further work.

## REFERENCES

- Abaidoo, V., & Arkorful, N. (2015). The role of e-learning, advantages, and disadvantages of its adoption in higher education. *International Journal of Instructional Technology and Distance Learning*, 2(12), 7.
- Agbonifo, O. C., & Olanrewaju, A. O. (2018). Genetic Algorithm-based Curriculum Sequencing Model For Personalized E-Learning System. *International Journal of Education and Management Engineering*, 5(8), 27–35.
- Alexandru, A., Tîrziu, E., Tudora, E., & Bica, O. (2015). Enhanced education by using intelligent agents in multi-agent adaptive e-Learning systems. *Studies in Informatics and Control*, 24(1), 13–22. <https://doi.org/10.24846/v24i1y201502>
- Alexandru, A., Tîrziu, E., Tudora, E., & Bica, O. (2016). Enhanced education by using intelligent agents in multi-agent adaptive e-Learning systems. *Studies in Informatics and Control*, 24(1), 13–22. <https://doi.org/10.24846/v24i1y201502>
- Balasubramanian, V., & Margret Anouncia, S. (2018). Learning style detection based on cognitive skills to support adaptive learning environment – A reinforcement approach. *Ain Shams Engineering Journal*, 9(4), 895–907. <https://doi.org/10.1016/j.asej.2016.04.012>
- Bergenti, F., Poggi, A., & Tomaiuolo, M. (2016). Multi-agent systems for e-health and telemedicine. In *Encyclopedia of e-Health and Telemedicine* (pp. 688–699). IGI Global.
- Bhansali, A. (2016). Architecture for improved adoption of eLearning in India. *Journal of Global Research in Computer Science*, 6(8), 1–4.
- Bokhari, M. U., & Ahmad, S. (2014). Detailed analysis of existing multi-agent-based E-learning

- systems. *2014 International Conference on Computing for Sustainable Global Development, INDIACom 2014*, 388–393. <https://doi.org/10.1109/IndiaCom.2014.6828165>
- Cakula, S., & Sedleniece, M. (2013). Development of a personalized e-learning model using methods of ontology. *Procedia Computer Science*, 26, 113–120.
- Chin, K. O., Gan, K. S., Alfred, R., Anthony, P., & Lukose, D. (2014). Agent architecture: An overview. *Transactions on science and technology*, 1(1), 18-35.
- Daoruang, B., Chaichomchuen, S., & Mingkhwan, A. (2018). *An Adaptive Learning System Based on Proportional VARK to Enhance Learning Achievement Concept. February*, 0–1.
- Dermeval, D., Paiva, R., Bittencourt, I. I., Vassileva, J., & Borges, D. (2018). Authoring tools for designing intelligent tutoring systems: a systematic review of the literature. *International Journal of Artificial Intelligence in Education*, 28(3), 336-384.
- Duo, S., & Ying, Z. C. (2012). Personalized e-learning system based on intelligent agents. *Physics Procedia*, 24, 1899–1902.
- El Fazazi, H., Elgarej, M., Qbadou, M., & Mansouri, K. (2021). Design of an Adaptive e-Learning System based on Multi-Agent Approach and Reinforcement Learning. *Engineering, Technology & Applied Science Research*, 11(1), 6637–6644. <https://doi.org/10.48084/etasr.3905>
- El Mhouthi, A., Nasseh, A., Erradi, M., & Vasquèz, J. M. (2017). Enhancing collaborative learning in Web 2.0-based e-learning systems: A design framework for building collaborative e-learning contents. *Education and Information Technologies*, 22(5), 2351–2364.

- Emelyanov, S., Makarov, D., Panov, A. I., & Yakovlev, K. (2016). Multilayer cognitive architecture for UAV control. *Cognitive Systems Research*, 39, 58–72.
- Fasihfar, Z., & Rokhsati, H. (2017). Study of the application of intelligent agents in e-learning systems. *Bull. La Société R. Des Sci. Liège*, 86 (February), 398–405.
- Gomes, P., Antunes, B., Rodrigues, L., Santos, A., Barbeira, J., & Carvalho, R. (2008). Using ontologies for e-learning personalization. *Communication & Cognition*, 41(1), 127.
- Hopgood, A. A. (2021). *Intelligent Systems for Engineers and Scientists: A Practical Guide to Artificial Intelligence*. CRC press.
- Hosni, H., El-Dosuky, M., & Eisa, M. (2020). Virtual learning environment and learning styles. *Journal of Theoretical and Applied Information Technology*, 98(17), 3595–3606.
- Howe, E., & Kekwaletswe, R. (2010). Utilizing Web 2.0 in a virtual learning environment: a South African context. *Proceedings of the IADIS Conference on E-Society*, 28–34.
- Hussain, I. (2017). Pedagogical implications of VARK model of learning. *Journal of Literature, Languages and Linguistics*, 38(2422-8435), 5.
- Kabassi, K., Dragonas, I., & Ntouzevic-Pilika, A. (2015). Learning management systems in higher education in Greece: Literature review. *2015 6th International Conference on Information, Intelligence, Systems, and Applications (IISA)*, 1–5.
- Kumar, A., Singh, N., & Ahuja, N. J. (2017). Learner characteristics based learning style models classification and its implications on teaching. *Int. J. Pure Appl. Math.(IJPAM)*, 118(20), 175–184.
- Lakkah, S. El, Alimam, M. A., & Seghiouer, H. (2017). Adaptive e-learning system based on

- learning style and ant colony optimization. *2017 Intelligent Systems and Computer Vision, ISCV 2017*. <https://doi.org/10.1109/ISACV.2017.8054963>
- Lhafiane, F., Elbyed, A., & Bouchoum, M. (2015). Multi-Agent System Architecture Oriented Prometheus Methodology Design for Reverse Logistics. *World Academy of Science, Engineering, and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9(8), 1914–1920.
- Maes, P. (2018). The agent network architecture (ANA). *Acm Sigart Bulletin*, 2(4), 115–120.
- Manate, B., Fortis, F., & Moore, P. (2014). Applying the Prometheus methodology for an Internet of Things architecture. *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, 435–442.
- Marciniak, J. (2014). Building intelligent tutoring systems immersed in repositories of e-learning content. *Procedia Computer Science*, 35, 541–550.
- McLeod, S. (2017). *Kolb's learning styles and experiential learning cycle*. *Simply psychology*, 5.
- Melesko, J., & Kurilovas, E. (2017). Personalized intelligent multi-agent learning system for engineering courses. *2016 IEEE 4th Workshop on Advances in Information, Electronic, and Electrical Engineering, AIEEE 2016 - Proceedings*, 1–6.  
<https://doi.org/10.1109/AIEEE.2016.7821821>
- Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing: A Review Journal*, 73, 1–15.  
<https://doi.org/10.1016/j.dsp.2017.10.011>
- Morandini, M., Dalpiaz, F., Nguyen, C. D., & Siena, A. (2014). The tropos software engineering

- methodology. In *Handbook on Agent-Oriented Design Processes* (pp. 463–490). Springer.
- Moubaidin, A., Shawarbeh, F., & Obeid, N. (2013). Using Intelligent Agents in e-Learning. *International Information Institute Tokyo*, 7325–7342.
- Nadrljanski, M., Vukic, D., & Nadrljanski, D. (2018). Multi-agent systems in e-learning. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings, October*, 990–995.  
<https://doi.org/10.23919/MIPRO.2018.8400181>
- Omicini, A. (2015). Event-based vs. multi-agent systems: towards a unified conceptual framework. *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 1–6.
- Oskouei, R. J., Varzeghani, H. N., & Samadyar, Z. (2014). Intelligent Agents: A Comprehensive Survey. *International Journal of Electronics Communication and Computer Engineering*, 5(August), 2278–4209.
- Padgham, L., & Winikoff, M. (2012). Prometheus: A methodology for developing intelligent agents. *International Workshop on Agent-Oriented Software Engineering*, 174–185.
- Pandey, A. (2017). *Benefits of Personalisation of eLearning*. <https://www.eidesign.net/top-8-benefits-personalisation-elearning/>
- Pour, B. S., Farajolahi, M., Sarmadi, M. R., & Shahsavari, H. (2017). Modeling of Personalized E-Learning Environment Based on Intelligent Agents. *Interdisciplinary Journal of Virtual Learning in Medical Sciences*, 8(3).
- Russell, S. J., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson.

<https://books.google.co.ke/books?id=B4xczgEACAAJ>

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson.

<https://books.google.co.ke/books?id=BDk7zgEACAAJ>

Saraswat, S. (2014). Customization and implementation of LMS moodle. *International Journal of Scientific and Research Publications*, 4(5), 1–4.

Saulnier, B. M., Landry, J. P., Longenecker Jr, H. E., & Wagner, T. A. (2008). From teaching to learning: Learner-centered teaching and assessment in information systems education. *Journal of Information Systems Education*, 19(2), 169.

Shehory, O., & Sturm, A. (2016). *AGENT-ORIENTED SOFTWARE ENGINEERING*. Springer.

Srivastava, B., & Haider, M. T. U. (2020). Personalized assessment model for alphabets learning with learning objects in e-learning environment for dyslexia. *Journal of King Saud University - Computer and Information Sciences*, 32(7), 809–817.

<https://doi.org/10.1016/j.jksuci.2017.11.005>

Szandała, T. (2021). Review and comparison of commonly used activation functions for deep neural networks. *Studies in Computational Intelligence*, 903, 203–224.

[https://doi.org/10.1007/978-981-15-5495-7\\_11](https://doi.org/10.1007/978-981-15-5495-7_11)

Suteja, B., & Istimewa, D. (2009). Implementing Ontology for E-learning System Development in Indonesia based on Web 2.0. *Int. J. Web Appl.*, 1(1), 29-46.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Wu, C. H., Chen, Y. S., & Chen, T. C. (2018). An Adaptive e-learning system for enhancing learning performance: Based on dynamic scaffolding theory. *Eurasia Journal of*

*Mathematics, Science and Technology Education*, 14(3), 903–913.

<https://doi.org/10.12973/ejmste/81061>

Xu, D., Huang, W. W., Wang, H., & Heales, J. (2014). *Enhancing e-learning effectiveness using an intelligent agent-supported personalized virtual learning environment: An empirical investigation*. Information & Management; Elsevier.

Zagulova, D., Boltunova, V., Katalnikova, S., Prokofyeva, N., & Synytsya, K. (2019).

Personalized E-Learning: Relation Between Felder– Silverman Model and Academic Performance. *Applied Computer Systems*, 24(1), 25–31. <https://doi.org/10.2478/acss-2019-0004>

Zajac, M. (2009). Using learning styles to personalize online learning. *Campus-Wide Information Systems*, 26(3), 256–265.

## APPENDICES

### APPENDIX I: The VARK Questionnaire

1. When learning from the Internet I like:

- Interesting design and visual features.
- Videos showing how to do or make things.
- Interesting written descriptions, lists and explanations.
- Audio channels where I can listen to podcasts or interviews.

2. A website has a video showing how to make a special graph or chart. There is a person speaking, some lists and words describing what to do and some diagrams. I would learn most from:

- Reading the words.
- Seeing the diagrams.
- Watching the actions.
- Listening.

3. I need to find the way to a shop that a friend has recommended. I would:

- Find out where the shop is in relation to somewhere I know.
- Use a map.
- Ask my friend to tell me the directions.
- Write down the street directions I need to remember.

4. I want to assemble a wooden table that came in parts (kitset). I would learn best from:

- Advice from someone who has done it before.
- Watching a video of a person assembling a similar table.
- Diagrams showing each stage of the assembly.
- Written instructions that came with the parts for the table.

5. I want to find out more about a tour that I am going on. I would:

- Talk with the person who planned the tour or others who are going on the tour.
- Look at details about the highlights and activities on the tour.
- Read about the tour on the itinerary.
- Use a map and see where the places are.

6. I prefer a presenter or a teacher who uses:

- Handouts, books, or readings.
- Question and answer, talk, group discussion, or guest speakers.
- Demonstrations, models or practical sessions.
- Diagrams, charts, maps or graphs.

7. I want to learn to do something new on a computer. I would:

- Follow the diagrams in a book.
- Start using it and learn by trial and error.
- Talk with people who know about the program.
- Read the written instructions that came with the program

8. I have a problem with my heart. I would prefer that the doctor:

- Used a plastic model to show me what was wrong.
- Showed me a diagram of what was wrong.
- Described what was wrong.
- Gave me something to read to explain what was wrong.

9. I want to learn how to take better photos. I would:

- Use the written instructions about what to do.
- Use diagrams showing the camera and what each part does.
- Use examples of good and poor photos showing how to improve them.
- Ask questions and talk about the camera and its features.

10. I want to learn about a new project. I would ask for:

- Diagrams to show the project stages with charts of benefits and costs.
- An opportunity to discuss the project.
- Examples where the project has been used successfully.
- A written report describing the main features of the project.

11. I want to save more money and to decide between a ranges of options. I would:

- Talk with an expert about the options.
- Read a print brochure that describes the options in detail.
- Use graphs showing different options for different times.
- Consider examples of each option using my financial information

12. When I am learning I:

- Use examples and applications.
- See patterns in things.
- Like to talk things through.
- Read books, articles, and handouts.

13. I have finished a competition or test and I would like some feedback. I would like to have feedback:

- From somebody who talks it through with me.
- Using a written description of my results.
- Using graphs showing what I achieved.
- Using examples from what I have done.

14. I want to learn how to play a new board game or card game. I would:

- Watch others play the game before joining in.
- Use the diagrams that explain the various stages, moves and strategies in the game.
- Read the instructions.
- Listen to somebody explaining it and ask questions.

15. When choosing a career or area of study, these are important for me:

- Using words well in written communications.
- Working with designs, maps or charts.
- Applying my knowledge in real situations.
- Communicating with others through discussion.

16. I want to find out about a house or an apartment. Before visiting it, I would want:

- A printed description of the rooms and features.
- To view a video of the property.
- A discussion with the owner.
- A plan showing the rooms and a map of the area.

## APPENDIX II: Source Codes

### Code for the tutor agent

```
<?php
class local_tutoragent_observer {
    /*public static function user_loggedin(\core\event\user_loggedin $event) {
        echo "<script>alert('logged in');</script>";    //call custom function
        $event_data = $event->get_data();
        var_dump($event_data);
        die();
    }
    public static function user_loggedout(\core\event\user_loggedout $event) {
        echo "<script>alert('logged out');</script>";    //call custom function
        $event_data = $event->get_data();
        var_dump($event_data);
        die();
    }
    public static function course_module_created(\core\event\course_module_created $event) {
        echo "<script>alert('Module Created');</script>";
        $event_data = $event->get_data();
        var_dump($event_data);
    }

    public static function content_viewed(\core\event\content_viewed $event) {
        echo "<script>alert('Content Viewed');</script>";
        $event_data = $event->get_data();
        var_dump($event_data);
        //get Vark value from DB
        $vark = 0;

        $table = $event->objecttable;
        $subjectid = $event->contextinstanceid;
        $courseid = $event->courseid;
        $userid = $event->userid;
        echo "Table: mdl_ ".$table."<br>Moduleid: ".$subjectid.<br>Courseid: ".$courseid."<br>Userid: ".$userid;
        $res = $DB->get_record($table, array(),
        $fields='name,intro', $strictness=IGNORE_MISSING);
        $intro = strip_tags($res->intro).'';
        $name = strip_tags($res->name).'';
        $input = $name.$intro.$vark;
        $recommender = new Recommender(__DIR__."/main.py");
```

```

        $response = $recommender->getRecommendation($input);
        // Pass response to the information agent
        \core\notification::info($response);
    }
}
?>

```

### Code for the information agent

```

<?php
class Recommender
{
    var $recommendation;
    function __construct($mainFile) {
        $this->recommendation = "C:\\Users\\Faijaz\\anaconda3\\envs\\tf\\python.exe". $mainFile;
    }
    function getRecommendation($text)
    {
        $text = $this->recommendation. " ". $text. "";
        $command = escapeshellcmd($text);
        $output = shell_exec($command);
        return $output;
    }
}
?>

```

## Code for the learner agent

```
import nltk
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()

import numpy
import tflearn
import tensorflow
import random
import json
import pickle

with open("intents.json") as file:
    data = json.load(file)

try:
    with open("data.pickle", "rb") as f:
        words, labels, training, output = pickle.load(f)
except:
    words = []
    labels = []
    docs_x = []
    docs_y = []

    for pattern in intent["patterns"]:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)
        docs_x.append(wrds)
        docs_y.append(intent["tag"])

    if intent["tag"] not in labels:
        labels.append(intent["tag"])

words = [stemmer.stem(w.lower()) for w in words if w != "?"]
words = sorted(list(set(words)))

labels = sorted(labels)

training = []
output = []

out_empty = [0 for _ in range(len(labels))]
```

```

for x, doc in enumerate(docs_x):
    bag = []

    wrds = [stemmer.stem(w.lower()) for w in doc]

    for w in words:
        if w in wrds:
            bag.append(1)
        else:
            bag.append(0)

    output_row = out_empty[:]
    output_row[labels.index(docs_y[x])] = 1

    training.append(bag)
    output.append(output_row)

training = numpy.array(training)
output = numpy.array(output)

with open("data.pickle", "wb") as f:
    pickle.dump((words, labels, training, output), f)
tensorflow.compat.v1.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)

    model.load("model.tflearn")
except:
    model.fit(training, output, n_epoch=1000, batch_size=8, show_metric=True)
    model.save("model.tflearn")
def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]

    s_words = nltk.word_tokenize(s)
    s_words = [stemmer.stem(word.lower()) for word in s_words]

```

```

s_words = nltk.word_tokenize(s)
s_words = [stemmer.stem(word.lower()) for word in s_words]

for se in s_words:
    for i, w in enumerate(words):
        if w == se:
            bag[i] = 1

return numpy.array(bag)

def access():
    print("Start to access the content")
    while True:
        inp = input("User: ")
        if inp.lower() == "quit":
            break

        results = model.predict([bag_of_words(inp, words)])
        results_index = numpy.argmax(results)
        tag = labels[results_index]

        for tg in data["intents"]:
            if tg['tag'] == tag:
                responses = tg['responses']

        print(random.choice(responses))
access()

```