Design of an Adaptive e-Learning System based on Multi-Agent Approach and Reinforcement Learning

Hanaa El Fazazi
Laboratory of Signals, Distributed Systems
and Artificial Intelligence
ENSET Mohammedia
University Hassan II
Casablanca, Morocco

Mouhcine Elgarej
Laboratory of Signals, Distributed Systems
and Artificial Intelligence
ENSET Mohammedia
University Hassan II
Casablanca, Morocco

Mohamed Qbadou
Laboratory of Signals, Distributed Systems
and Artificial Intelligence
ENSET Mohammedia
University Hassan II
Casablanca, Morocco

Khalifa Mansouri
Laboratory of Signals, Distributed Systems
and Artificial Intelligence
ENSET Mohammedia
University Hassan II
Casablanca, Morocco

Abstract-Adaptive e-learning systems are created to facilitate the learning process. These systems are able to suggest the student the most suitable pedagogical strategy and to extract the information and characteristics of the learners. A multi-agent system is a collection of organized and independent agents that communicate with each other to resolve a problem or complete a well-defined objective. These agents are always in communication and they can be homogeneous or heterogeneous and may or may not have common objectives. The application of the multi-agent approach in adaptive e-learning systems can enhance the learning process quality by customizing the contents to students' needs. The agents in these systems collaborate to provide a personalized learning experience. In this paper, a design of an adaptative e-learning system based on a multi-agent approach and reinforcement learning is presented. The main objective of this system is the recommendation to the students of a learning path that meets their characteristics and preferences using the Q-learning algorithm. The proposed system is focused on three principal characteristics, the learning style according to the Felder-Silverman learning style model, the knowledge level, and the student's possible disabilities. Three types of disabilities were taken into account, namely hearing impairments, visual impairments, and dyslexia. The system will be able to provide the students with a sequence of learning objects that matches their profiles for a personalized learning experience.

Keywords-adaptative e-learning system; knowledge level; learning path recommendation; learning styles; multi-agent; Q-learning; reinforcement learning; students' disabilities

I. Introduction

Intelligent tutoring systems offer enable students to get educational resources that satisfy their requirements or interests [1]. Most of the methods applied for adaptation are based on the individual characteristics of the learner in order to produce personalized learning content. The most used characters and preferences in adaptive e-learning environments are knowledge level and learning style [2]. The best known and most used is the Felder-Silverman Learning Style Model (FSLSM) [3]. This model is based on eight dimensions, each one containing two categories [4]. Authors in [5] conducted a comparative study and analyzed the existing e-learning systems that use the multi-agent approach. The conclusion was that none of the studied systems is open source and they all lack in including all the desired characteristics (collaborativeness, flexibility, adaptability, interactivity, and security) in the same system. Authors in [6] proposed a model and architecture for developing a personalized multi-agent e-learning system. Authors in [7] developed a model of an intelligent learning system based on an agent approach. This model is inspired by intelligent tutoring to present adaptability to distributed virtual learning environments. In addition of using multi-agent, the application of machine learning techniques can also improve e-learning systems and make them more and more intelligent [8]. Reinforcement learning, as a field of machine learning, can help students to obtain appropriate teaching materials from the learning process. Authors in [9] give a technique based on reinforcement learning, more specifically the Q-learning algorithm, to produce an adaptive system that matches the needs of each student. Authors in [10] suggest an online learning method based on reinforcement learning that can deal with complex environments. The used algorithm, chooses suitable learning objects to recommend to a given student.

The literature review showed a lack in the development of e-learning solutions that respond to various characteristics of students. Most systems take into account one or two characteristics of learners and students' disabilities are little considerate in these solutions. We have tried to propose a system that takes into consideration the learning styles, knowledge level, and disabilities of the students by applying the multi-agent approach and reinforcement learning to provide an intelligent and adaptive e-learning system. This system will

Corresponding author: Hanaa El Fazazi

be able to generate the learning path to each student based on individual needs and preferences.

In this paper, a design of an adaptive e-learning system based on a multi-agent approach and reinforcement learning is presented. The system will be able to recommend a learning path to students based on their learning style, knowledge level, and disabilities. Three types of disabilities were taken into consideration, namely hearing impairment, visual impairment, and dyslexia. The application of multi-agent approach and reinforcement learning will increase system performance and make the process of adaptation more flexible and provide the underlying power of the proposed model.

II. The Proposed Adaptive e-Learning System

The proposed adaptive e-learning system aims to recommend a personalized, useful, and interesting learning resource based on students' characteristics and preferences. The principal components used in this system are the Learner Model, the Content Model, and the Adaptation Model. These components interact with each other to provide a customized adaptation of the system according to the Learner Model, which gathers all the information about the student, and the Content Model, which describes the learning resources and their structure. This system will try to choose the most suitable content structure for each student, taking into consideration his/her learning style, knowledge level, and disabilities.

A. Content Model

The content model contains the learning resources structured in a way that facilitates the process of adaptation when using intelligent e-learning systems. The adopted model consists of four layers represented as a hierarchical network of four levels: Course, Chapter, Learning Unit, and Learning Object (LO) as shown in Figure I 1.

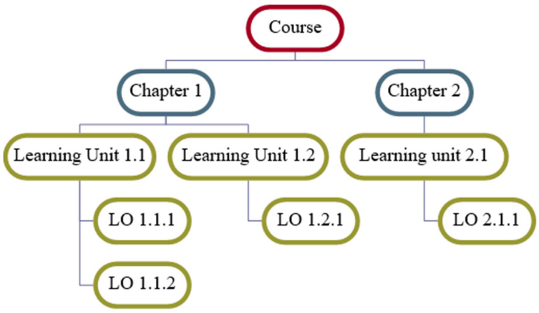

Fig. 1. Course organization.

The "Course" layer is located at the top of the network and represents the root of the content model structure. In this context, the course is composed of a series of chapters corresponding to the different teaching concepts on a particular topic. Below the course root, there are some "Chapters", each of which deals exclusively with a particular element of the course. The third layer, the "Learning Unit" layer, contains the different learning units that deal with a part of the course (chapter), this learning unit at a level that is classified primarily as beginner, intermediate or advanced, each category determines its importance to the learner according to the knowledge level. The fourth layer contains the LOs that are associated with the learning unit. The learning object represents the smallest element of the proposed decomposition. Each element of the given structure should be well indexed for the next goal of searching and reusing the learning material. For that reason, all of the elements in the designed structure can also add metadata, which is information about the component itself. These metadata can include the title, author, education level, level of difficulties, interactivity level and type, etc. By applying metadata, learning material searching and recognizing becomes easier since it can contain various information that can be used as an identity for a certain LO. In the context of the defined ontology methods, Table I reviews the metadata that can be used for each element described in Figure 1.

TABLE I. COURSE COMPONENT METADATA

Component	Metadata
Course	name, dificulty level, course description, course language, course duration
Chapter	name, chapter level, prerequisite, chapter description
Learning Unit	name, goal, prerequisite, unit level,
LOs	ID-LO, name, author, creation date, prerequisite, last modified, difficulty level, keyword, level of interactivity, disabily type, format type, relation type

B. Learner Model

Adaptive systems typically use different user data depending on the purpose of the adaptation. In our approach, the user is a student, so the system should be able to customize the learning experience according to the needs of the students. Since there is a wide range of learner characteristics that can be used in the Learner Model, the selection of the appropriate characteristics is necessary. A Learner Model is considered complete if it includes learning history, learner progress in the current learning element, learning style, and other types of learner information [11]. The model we used in the proposed system is based on five dimensions:

- General personal information like name, age, gender, etc. generally remains static throughout the use of the system.
- Personal preferences like the preferred domain, language and media type.
- The disability dimension is represented by three types of disabilities, namely hearing impairment, visual impairment, and dyslexia. These disabilities are described with a degree which refers to the gravity of the disabilities and it takes as values: Low, Medium, and Severe.
- Learning styles are defined according to the Felder-Silverman model [4].
- Information on personal learning status, which brings together the knowledge level, history, and learning objectives to generate a customize adaptation. This type of information will be constantly updated during the learning process.

Figure 2 describes the ontology used to define these dimensions. The learner profile explains how the learner learns best. It is the standard representation of learners' data that can be collected in two ways: directly from the student or by analyzing his performance through a learning management system [11]. In the first interaction of the learner with the system and after the provision of personal information and type of disability, the learner takes the ILS questionnaire to identify the learning style according to FSLSM [12]. Then, and after he/she chooses the course to study, the learner is asked to take an initial test to define his/her knowledge level and then build the learner profile. All this information is managed by the Learner Agent who will be explained in detail below.

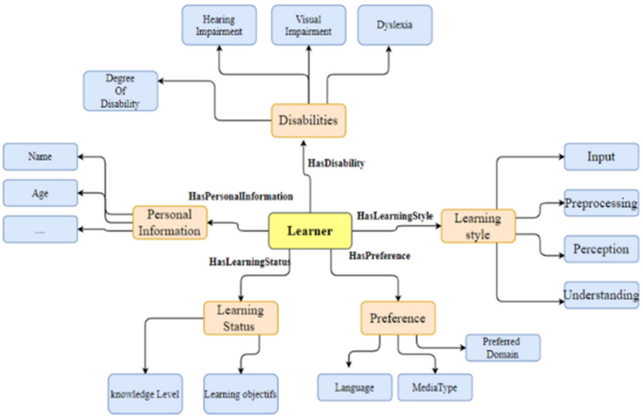

Fig. 2. Learner ontology.

C. *Adaptation Model*

The Adaptation Model aims to generate a suitable learning path in order to enhance learning. It uses the information stored in the Learner Model and the Content Model to provide the appropriate recommendation. The Adaptation Model provides a sequence of LOs according to the learning style, knowledge level, and disabilities of the student. This approach is based on the Q-Learning algorithm to decide which learning path to take in order to increase the results of the learning process. As presented in the Content Model and by choosing a learning unit, the learner can browse through the different LOs of the chosen learning unit. The learning unit can contain theoretical files, examples, exercises, quizzes, diagrams, etc. The learner can browse through these elements (by reading a theoretical file, seeing an example, solving an exercise, taking a quiz). The objective of our system is to choose the LOs and actions to be done for a more enriching learning process according to the learner profile. For this, we chose to use the Q-Learning algorithm, which is based on states and actions. The states in our approach are the LOs and the actions are the relations that link them. Each action has a well-defined reward and the algorithm will choose the path with the maximum rewards. The recommendation task is provided by the Adaptation Agent, who will be discussed below.

III. Multi-Agent System Architecture

In this paper, a distributed architecture based on the multi-agent approach for modeling the communication and coordination process is proposed which allows the suggested e-learning system to recommend LOs adapted to the learner's profile according to his/her characteristics and preferences. To design this distributed model, we used a set of autonomous agents that are provided with behaviors (tasks and actions to be performed) to search and design an adequate response for the learners. This architecture is interconnected with two systems (two platforms), namely a multi-agent platform designed with Java and the Moodle LMS system which is designed with PHP. To keep the communication and coordination process between these two systems a set of web services was adopted along with a shared database between them to facilitate access to information and keep the system running easily [13]. Figure 3 shows the adopted multi-agent architecture for the implementation of the proposed models to adapt LOs. This architecture is based on intelligent agents, each agent is characterized with a set of behaviors and roles, i.e. the way to search for data and the way to process and adapt the results according to the requests of other agents, without forgetting the aspect of each agent's autonomy. These agents interact with each other to provide customized content adapted to the characteristics of the students' profile. In our case, each agent is equipped with a knowledge base represented by scenarios to be followed when the system meets the same requests.

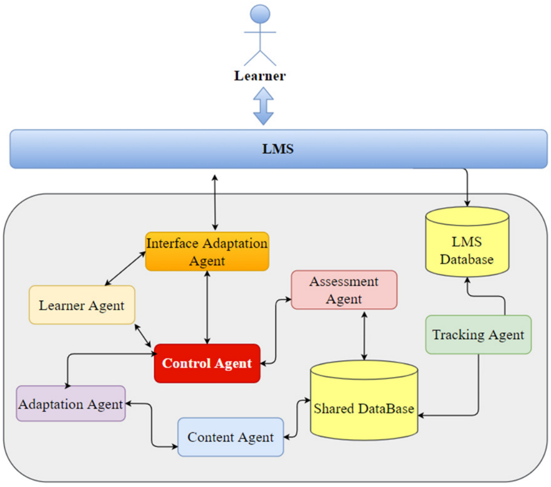

Fig. 3. The multi-agent architecture of the proposed system.

Each agent entity will be modeled by a sequence of roles and tasks (Figure 4) to be executed according to the needs of our system. The coordination process is executed dynamically, each agent can request information or data from other agents. The meshed topology gives us the possibility to have a highly connected architecture. This structure allows the system to optimize the inter-agent communication process as well as to speed up the work rate of the agents to build the best results in a short time. The communication process between agents is characterized by a communication protocol defined by the development platform used. To open a communication sequence between two agents, it is necessary to start first with the authentication of agents to download the profile of each entity. Then a communication channel will be reserved for the sharing of data and actions according to a set of commands (INFORM, REQUEST, RESPONSE). All communication process will be saved in a database that will be linked with the LMS database. This link intends to separate the structural architecture of the Moodle system from our agent-oriented architecture (Figure 5), without forgetting that our ontological architecture is also adapted and transformed into the basic structure of the Moodle system to facilitate the process of injecting and consulting data. For data synchronization between the two databases, a set of triggers was used that are provided by the database management system to modify the content of each table without going through the GUI.

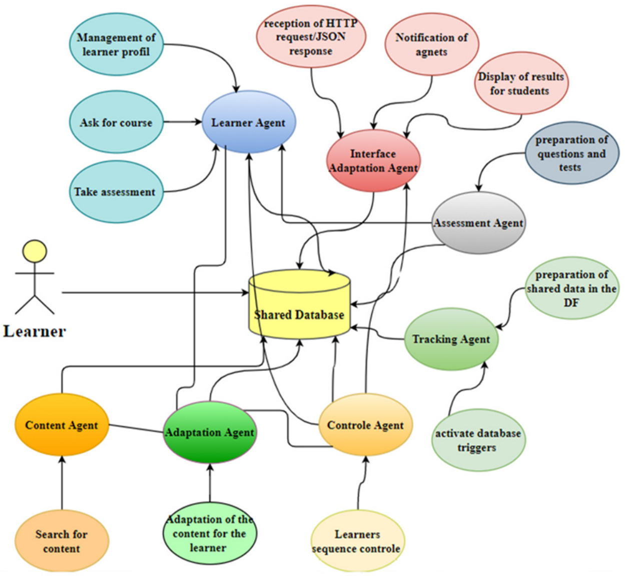
Management of learner profil
Ask for course
Take assessment
Learner Agent
reception of HTTP request/JSON response
Notification of agnets
Display of results for students
Interface Adaptation Agent
preparation of questions and tests
Assessment Agent
Learner
Shared Database
preparation of shared data in the DF
Tracking Agent
Content Agent
Adaptation Agent
Controle Agent
activate database triggers
Search for content
Adaptation of the content for the learner
Learners sequence controle

Fig. 4. Role and task model of agents.

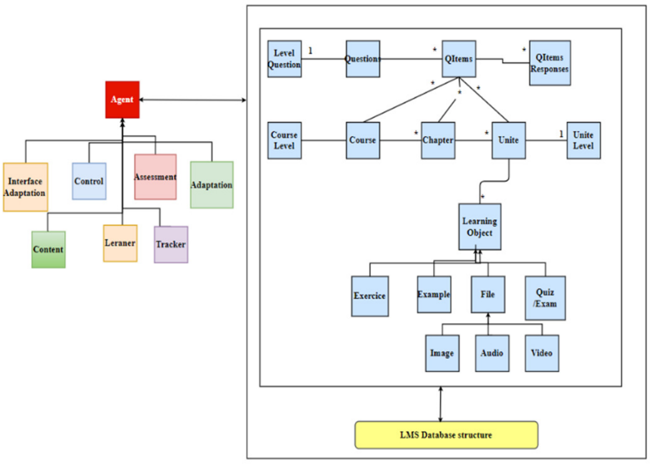

Fig. 5. Agent-oriented conceptual model for the interactions between the multi-agent system platform and the LMS.

IV. Role and Task Model Description for E-Learning System Actors

The modeled architecture is based on multiple agents. These agents are designed to communicate and exchange data in order to provide relevant information to students, such as adequate learning resources that match their characteristics and preferences. The agents in this system must collaborate to carry out the process of adapting LOs. In our architecture, there are two types of agents. The first type is instantiated once and is a single entity that will be available to perform all the computation and processing required meeting the agents' requests. This approach allows us to reduce the number of agents and to centralize the calculation process in entities that have the roles and calculation methods, to prepare in advance the input values that will be reused by the other agents. The other type of agent (e.g. the Learner Agent) will be linked to students who are searching for courses in our system. This approach will allow us to create for each student an instance of the Learner Agent to control his/her profile and give our system the chance to distribute the calculation process and share the information among the students. This mechanism will allow us to make a decomposition of the global system with sub-systems to better manage the life cycle of each learner.

A. Adaptation Interface Agent

The Adaptation Interface Agent manages all communications between the interface and the system (Figure 6). This agent acts as an intermediary who communicates the student's requests to the appropriate agents and then presents the learning material in a suitable format for the student. For the implementation of this agent, we opted to modify the source code of the Moodle LMS in order to integrate our own HOOK Manager. These elements are needed to interpret requests from students and users of the Moodle system to adapt it to our multi-agent system. In a multi-agent architecture, we need to have triggers, the role of these triggers is to inform/send a request or query to an agent to perform a task according to the parameters injected into the query. To enable this communication, the system will need an interface between two different types of programming language, namely the object-oriented PHP with which the Moodle system is developed, and the object-oriented Java used for setting up standalone agents. Interfacing between these agents is done with web services. The role of these web services is to send HTTP requests between the two platforms in order to design a communication middleware or a communication bridge between the two systems.

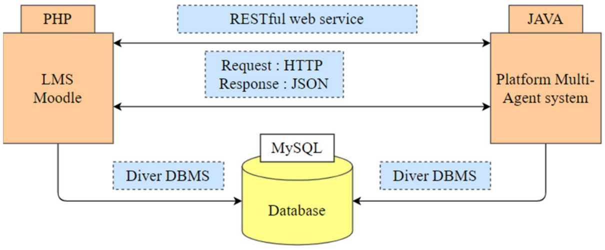

Fig. 6. Communication model between the used platforms.

B. Learner Agent

In our system, each student will have his own Learning Agent, the purpose of this multiplicity is to create a distributed system where each student has his own preferences, characteristics, and pedagogical orientations. As soon as a student is logged in the LMS, an agent will be created to encapsulate that learner's personal data. It should be noted that in order to have a communication channel with the other agents, we must go through this agent. The Learner Agent is responsible for collecting information and storing and managing the data in the student's profile. This agent is in periodic communication with a set of agents namely the Adaptation Interface agent and the Control Agent. This agent is characterized by two main roles:

- Knowledge level calculation: each student must take an initial test for a specific course. The results of this test will be processed to identify the student's knowledge level according to the category "beginner", "intermediate", or "advanced".
- Learning style calculation: the student has to take the ILS questionnaire. This questionnaire contains 44 questions [11]. The Learner Agent will process the answers provided by the learner and calculate the learning style according to the FSLSM model. The learning style will be used by our agents to prepare the learning content according to the characteristics and needs of the learner.

C. *Control Agent*

The role of the Control Agent is to manage the calculation process of the agents and facilitate the access to information to speed up the calculation time within the system. This agent keeps the synchronization process between the agent-oriented system and the Moodle platform. This agent controls the agent life cycle. Sometimes the system can have anomalies in the execution of the agents or an overload processing may occur in the system, something usual when using java. So, we thought of an automated solution that allows us to create a set of Cron Jobs (scheduled tasks) that will be executed in the background to empty the agent cache and to restart the agents in a blocking state or else destroy the agent and restart it again.

D. *Evaluation Agent*

Each course and unit within our system is characterized by an evaluation that helps us to position the learner in his learning process. In our case we have three different types of evaluation, namely initial tests, post-tests, and exams. This agent can load the list of the assessments associated with each course/unit to evaluate the student's knowledge level at each stage of the learning process. This process is required in order to evaluate the student's learning level and see if the suggested LOs by the system have improved it.

E. *Adaptation Agent*

The Adaptation Agent is responsible for generating the learning path to the learner according to the data received from the Control Agent and the Learner Agent. The Adaptation Agent communicates with the Content Agent to request the LOs that correspond to the student's learning style, knowledge level, and disabilities. Once these data are received, the Adaptation Agent applies the Q-Learning algorithm to choose the most beneficial learning path for the current learner. The learning path will then be stored in the database. The process can be repeated if the learning style or one of the adaptation elements is modified. The Adaptation Agent records the results of the adaptation so that it can be reused for students with similar characteristics.

F. *Content Agent*

The Content Agent is responsible for searching for LOs in the database according to the query sent by the Adaptation Agent. It looks for LOs that correspond to the learning styles, knowledge level, and disabilities of the learner for a given learning unit. For each unit we have identified a list of LOs, so the Content Agent will launch a query to select and retrieve the list of LOs with a filter optimized according to the learner's profile. The hierarchical structure of the course entity facilitates the process of searching and constituting adequate resources for the learner. The content selected by this agent will then be presented to the learner. This content will then be saved in the database. If this agent will receive the same profile, he will just consult this table to retrieve the content, thus we can reduce the time wasted in processing requests.

G. *Tracking Agent*

As mentioned before, our system is a hybrid platform consisting of two modules. A module for course management and construction, which will be controlled by the Moodle LMS, and another which is dedicated to the multi-agent architecture. To keep the two modules synchronized we have designed a synchronization mechanism to adapt the content and data already stored in Moodle to the needs of our agents. To perform this task, we have designed triggers at the database level and a set of Hooks to encapsulate student HTTP requests. The role of the Tracking Agent is to detect all new events in our system. It will use these Hooks to filter the requests from the students and to instantiate and trigger the appropriate agents.

V. SIMULATION OF THE CONTENT ADAPTATION PROCESS

As the learning process is initialized in the system, each agent goes through a series of steps that form its life cycle. Agents must communicate with each other in order to be able to exchange and request actions. To describe this communication process, we have used sequence diagrams to represent the interactions between the different agents of the proposed system. During the first interaction with the system, the learner is required to log in and fill in a personal information form. The next step is to take the ILS questionnaire to identify the learning style according to FSLSM. Once the learner's profile is finalized, the learner can start using the system and choose the courses and learning elements that he or she wants to learn. At the beginning of the learning experience, the learner must identify the course to be learned. At this stage, the system asks the learner to take the initial test to determine his/her knowledge level. The initial test includes various concepts that the learner must understand. These concepts are represented by questions from each learning unit of the selected course with different levels of difficulty. The obtained score in this initial test allows us to classify the student in one of the three categories, i.e. "beginner", "intermediate", or "advanced". After determining the knowledge level, the student can begin the learning process by choosing the units to be learned. The system then presents the LOs of this unit and adapts them to the learner's profile.

Figure 7 shows the first interaction between a learner and the system. After the authentication of the learner, the system will instantiate a Learner Agent to manage the profile of this student. Then the system will propose the ILS questionnaire. After the student completes the questionnaire, the Learner Agent will be able to calculate the ILS score and identify the learner's learning style according to the FSLSM model. The identified learning style will be added to the student's profile in the database, and will be displayed for the student. To identify the student's knowledge level about a specific course, three agents are used: The Interface Adaptation Agent, the Learner Agent and the Assessment Agent. The course chosen by the student is transmitted to the Learning Agent. This message will be communicated to the Evaluation Agent to prepare the initial test by choosing a set of relevant questions from the database with different degrees of difficulty. After passing the initial test, the Learner Agent calculates the obtained score. This computation is carried out according to the classic method because each question consists of an answer indexed by a key, so when the student ticks an answer, the system will retrieve the value of the selected answer. The final result of this test will be used to identify the student's knowledge level which will be added to the learner profile to be used in learning scenarios.

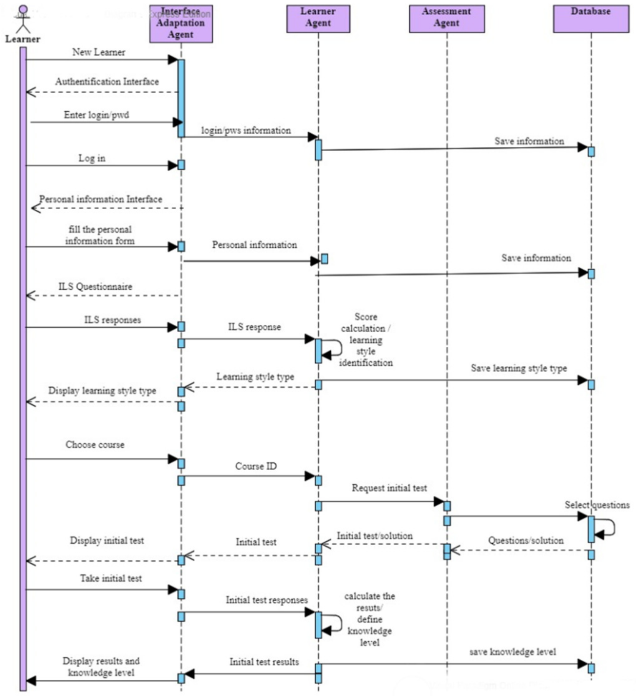

Fig. 7. Sequence diagram of the learner's first interaction with the system.

Figure 8 explains the adaptation process. After receiving the parameters from the Learner and Control Agents, the Adaptation Agent charges the Content Agent to select the LOs according to the learning style, the knowledge level, and disabilities of the learner and regarding a well-defined learning unit. Once all LOs have been retrieved, they will then be transformed to the Adaptation Agent to determine the order in which they are displayed. Because according to the learner's learning style, the system will reorder the LOs in a logical order that corresponds to the learner. At the end of this step, the Learner Agent will record the content exposed for the learner in his profile and then this data can be applied and reused in other cases if the system meets a learner who has the same characteristics as the current student.

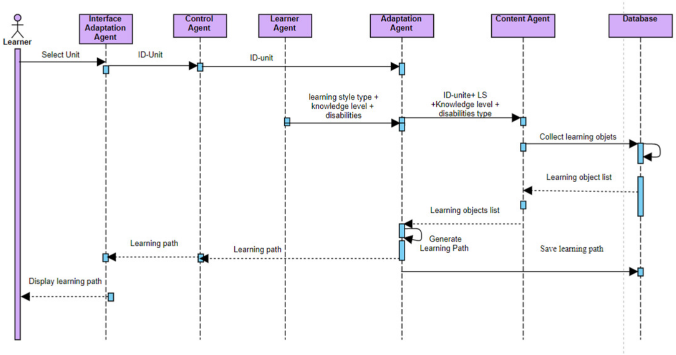

Fig. 8. Sequence diagram of adaptation process.

VI. IMPLEMENTATION EXAMPLE FOR LEARNING PATH RECOMMENDATION

In this example, we assume the case of a learner who has the following profile: intermediate knowledge level, verbal learning style, and a hearing impairment. The Adaptation Agent receives this information and then communicates it to the Content Agent to select the LOs and their relationships. As discussed above, the LOs represent states and the relationships between them represent actions. The Content Agent returns all the resources that match the learner's profile. The Adaptation Agent must choose the most beneficial path for the current learner. In this example, the Content Agent reviews 5 learning objects: two text files, an example, an exercise, and a final test with the following actions: ReadFile, ReadMore, SolveExercise, SeeExample, TakeFinalTest (Figure 9). Each action has a reward:

- ReadFile: +50
- ReadMore: +50
- SolveExercice: +40
- SeeExample: +20
- TakeFinalTest: +30
- Previous: +10

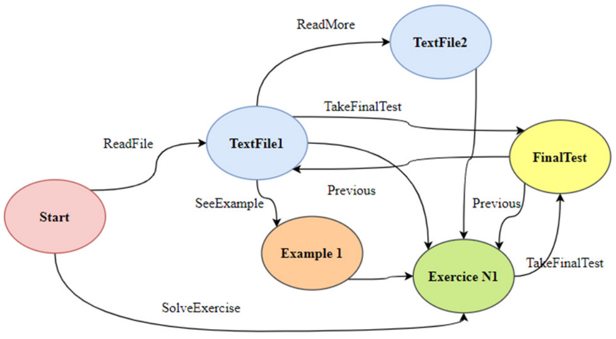

Fig. 9. Example of LO list for the given student.

TABLE II. REWARD TABLE OF ASSUMED STATE-ACTION COMBINATION

	0	**1**	**2**	**3**	**4**	**5**
0	0	+50	0	0	+40	0
1	0	0	+50	+20	+40	+30
2	0	0	0	0	+40	+30
3	0	0	0	0	+40	0
4	0	0	0	0	0	+30
5	0	+10	0	0	+10	0

TABLE III. Q-TABLE FOR THE ASSUMED STATE-ACTION COMBINATION

	0	**1**	**2**	**3**	**4**	**5**
0	0	115.7229998544553	0	0	60.999999999991	0
1	0	0	93.8899999588963	62.699999971061	60.999999955544	29.9999999997890
2	0	0	0	62.6999999999999	0	29.9999999999999
3	0	0	0	0	60.999999999999	0
4	0	0	0	0	0	29.9999999999999
5	0	0	0	0	0	0

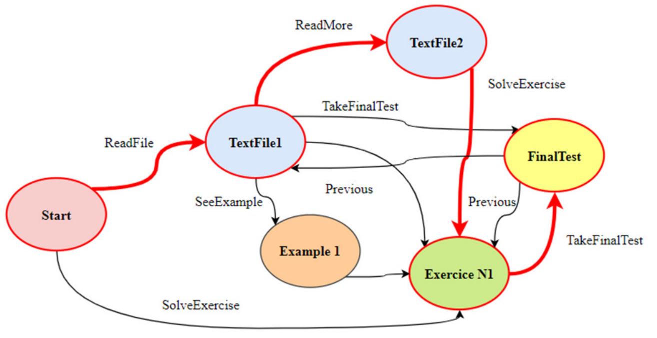

Fig. 10. Example of a learning path starting from the beginning.

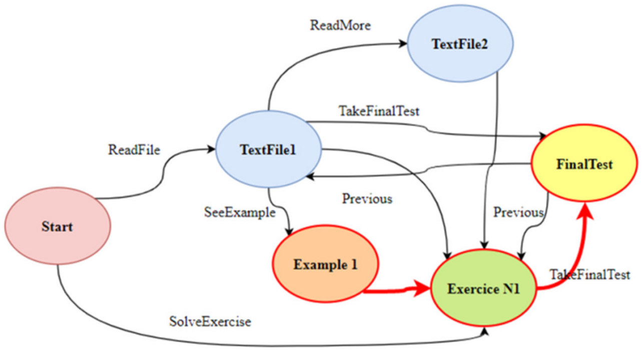

Fig. 11. Example of a learning path starting from an example.

The rows of Table II represent the states and the columns represent the actions. The result of the Q-Learning algorithm application to the above example are shown in Table III. The algorithm proposes the best learning path for a given student. The learner can choose to start with any LO and the system will provide the best learning path. For example, if the learner chooses to start learning from the beginning, the best learning path in this case is 0-1-2-4-5 (Figure 10). If the student chooses to start with an example, the best learning path is 3-4-5 (Figure 11).

VII. Conclusion

In this paper, a design for an adaptive e-learning system based on a multi-agent approach and reinforcement learning has been proposed. The architecture used in this system is based on many agents, each agent taking charge of a well-defined task. The agents are all connected and they communicate permanently to ensure the regular operation of the system. The main functionality proposed in this system is the recommendation of learning paths according to the learning styles, the knowledge level, and the disabilities of the learners using the Q-Learning algorithm. The proposed architecture provides personalization of the learning experience and provides LOs that match the profile and the needs of the students. It is a distributed architecture based on autonomous agents that constantly communicate to respond to learner requests. The proposed system can be integrated with any learning management system. The functionalities adopted in this system allow it to interact and communicate with other systems without any obstacle. This interoperability is ensured through the use of web services as well as the ontological representation of the data. In future work, we intend to consider more states and actions to find the optimal learning path. This feature can increase the complexity of the model. Deep Q-Learning seems to be a good solution to this issue.

References

[1] M. Laaziri, S. Khoulji, K. Benmoussa, and K. M. Larbi, "Outlining an Intelligent Tutoring System for a University Cooperation Information System," *Engineering, Technology & Applied Science Research*, vol. 8, no. 5, pp. 3427–3431, Oct. 2018, https://doi.org/10.48084/etasr.2158.

[2] M. T. Alshammari and A. Qtaish, "Effective Adaptive E-Learning Systems According to Learning Style and Knowledge Level," *Journal of Information Technology Education: Research*, vol. 18, pp. 529–547, Nov. 2019.

[3] M. Abdullah, W. H. Daffa, R. M. Bashmail, M. Alzahrani, and M. Sadik, "The Impact of Learning Styles on Learner's Performance in E-Learning Environment," *International Journal of Advanced Computer Science and Applications (ijacsa)*, vol. 6, no. 9, 45/01 2015, https://doi.org/10.14569/IJACSA.2015.060903.

[4] R. Felder, "Learning and Teaching Styles in Engineering Education," *Journal of Engineering Education -Washington-*, vol. 78, no. 7, pp. 674–681, Jan. 1988.

[5] M. U. Bokhari and S. Ahmad, "Multi-Agent Based E-Learning Systems: A Comparative Study," in *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*, New York, NY, USA, Oct. 2014, pp. 1–6, https://doi.org/10.1145/2677855.2677875.

[6] P. Q. Dung and A. M. Florea, "An Architecture and a Domain Ontology for Personalized Multi-agent e-Learning Systems," in *2011 Third International Conference on Knowledge and Systems Engineering*, Hanoi, Vietnam, Oct. 2011, pp. 181–185, https://doi.org/10.1109/KSE.2011.35.

[7] C. Giuffra and R. Silveria, "A multi-agent system model to integrate Virtual Learning Environments and Intelligent Tutoring Systems," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 2, no. Special Issue on Artificial Intelligence and Social Application, 2013.

[8] M. Krendzelak, "Machine learning and its applications in e-learning systems," in *2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Stary

Smokovec, Slovakia, Dec. 2014, pp. 267–269, https://doi.org/10.1109/ICETA.2014.7107596.

[9] M. Boussakssou, B. Hssina, and M. Erittali, "Towards an Adaptive E-learning System Based on Q-Learning Algorithm," *Procedia Computer Science*, vol. 170, pp. 1198–1203, Jan. 2020, https://doi.org/10.1016/j.procs.2020.03.028.

[10] W. Intayoad, C. Kamyod, and P. Temdee, "Reinforcement Learning for Online Learning Recommendation System," in *2018 Global Wireless Summit (GWS)*, Chiang Rai, Thailand, Nov. 2018, pp. 167–170, https://doi.org/10.1109/GWS.2018.8686513.

[11] K. R. Premlatha and T. V. Geetha, "Learning content design and learner adaptation for adaptive e-learning environment: a survey," *Artificial Intelligence Review*, vol. 44, no. 4, pp. 443–465, Dec. 2015, https://doi.org/10.1007/s10462-015-9432-z.

[12] "Index of Learning Styles Questionnaire," *NC State University.* https://www.webtools.ncsu.edu/learningstyles/ (accessed Dec. 20, 2020).

[13] N. E. A. Amrani, O. E. K. Abra, M. Youssfi, and O. Bouattane, "A Novel Deep Learning Approach for Semantic Interoperability Between Heteregeneous Multi-Agent Systems," *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, pp. 4566–4573, Aug. 2019, https://doi.org/10.48084/etasr.2841.