A Multi-agent Based Adaptive E-Learning System

Birol Ciloglugil[1(✉)], Oylum Alatli[1], Mustafa Murat Inceoglu[2], and Riza Cenk Erdur[1]

[1] Department of Computer Engineering, Ege University, 35100 Bornova, Izmir, Turkey
{birol.ciloglugil,oylum.alatli,cenk.erdur}@ege.edu.tr

[2] Department of Computer Education and Instructional Technology, Ege University, 35100 Bornova, Izmir, Turkey
mustafa.inceoglu@ege.edu.tr

Abstract. In this paper, a multi-agent based adaptive e-learning system that supports personalization based on learning styles is proposed. Considering that the importance of distance education has increased with the effect of the Covid-19 pandemic, it is aimed to propose an adaptive e-learning system solution that offers more effective learning experiences by taking into account the individual differences in the learning processes of the students. The Felder and Silverman learning style model was used to represent individual differences in students' learning processes. In our system, it is aimed to recommend learning materials that are suitable for learning styles and previous knowledge levels of the students. With the multi-agent based structure, an effective control mechanism is devised to monitor the interaction of students with the system and to observe the learning levels of each student. The purpose of this control mechanism is to provide a higher efficiency in the subjects the students study compared to non-personalized e-learning systems. This study focuses on the proposed architecture and the development of the first prototype of it. In order to test the effectiveness of the system, personalized course materials should be prepared according to the learning styles of the students. In this context, it is planned to use the proposed system in future studies within the scope of a course in which the educational content is personalized.

Keywords: Adaptive E-Learning · Personalization · Multi-agent system · Felder and Silverman learning style model

1 Introduction

E-Learning systems have been used widely for decades; however, providing effective e-learning systems to students has gained significant importance especially after the Covid-19 pandemic [1–3]. Both educators and students have faced problems such as lack of interaction, loss of motivation and less effective learning experiences during this difficult period, which resulted in a significant increase

O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12951, pp. 693–707, 2021.
https://doi.org/10.1007/978-3-030-86970-0_48

in adaptive e-learning systems research among different kinds of e-learning systems research being conducted.

Adaptive e-learning systems provide personalized solutions by modeling individual differences of students in order to provide them personalized learning experiences [4]. Students may have different learning objectives, preferences, background knowledge levels, individual needs that can affect the way they learn [5,6]. Learning style models are widely used for modeling these individual differences in adaptive e-learning systems [7–10]. In this study, in order to model individual differences in students' learning processes, we used Felder and Silverman learning style model (FSLSM) [11], which is the most-commonly used model in the literature. In addition to FSLSM, previous knowledge levels of the students are also used as a source to provide personalized learning materials to them.

Software agents are a good choice for the implementation of adaptive e-learning systems [12]. Software agents have been one of the widely-used technologies in various fields and industries such as industrial engineering, economics and financial management, medical science, traffic and transportation, defense, business and management [13]. They have also been utilized in the e-learning field to provide multi-agent based e-learning systems [14–23].

Software agents technology is frequently used with Semantic Web technology [14,24]. As an important component of the Semantic Web, ontologies are used to develop data representation models in various domains [25]. In the e-learning field, ontologies are used for modeling students and learning materials [26]. These ontologies can be used in different e-learning systems, and therefore, increase interoperability of them [26]. Learning objects and learning style models have been modeled with ontologies in different studies [4,5,15–17,22,27–30]. E-learning systems that support both multi-agent and Semantic Web technologies include [15,16,22,31,32].

In this paper, we propose a multi-agent based adaptive e-learning system that utilizes ontological representation of Felder and Silverman learning style model as part of the student model. Learning materials provided by the lecturers are also annotated ontologically. Therefore, students can access learning materials personalized according to their learning styles by using the proposed e-learning system. Multi-agent based components of the system provide the interface for both the students and the lecturers using the system. The multi-agent features also support an effective control mechanism to monitor interaction of the students with the system and to observe the learning levels of each student.

The rest of the paper is organized as follows; Sect. 2 briefly introduces the software agent notion and usage of multi-agent systems in different domains. Then, the related work of multi-agent systems in e-learning domain is discussed. Section 3 presents the architecture and the prototype implementation of the proposed multi-agent based adaptive e-learning system. Finally, Sect. 4 concludes the paper with the future work directions intended for the first prototype.

2 Related Work

In this section, first, we will introduce software agents and multi-agent systems briefly. Then, we will examine how multi-agent systems are used in the e-learning domain by discussing various studies.

2.1 Multi-agent Systems

There are different definitions for the term agent. According to the most fundamental definition [33], an agent is a system that perceives its environment using its sensors and interact with its environment using its effectors. In order to classify a program as an agent there are some essential requirements that the program should have [34]: autonomy, social-ability, reactivity and pro-activeness. Autonomy means, programs defined as agents should be able to operate on their own without any intervention. Social ability indicates the necessity of being able to interact with other agents and people via an agent communication language. Reactivity requires the agent to sense its environment and respond to the changes in it when necessary. However, reactivity is not enough. Agents should be able to behave in a goal-directed manner and take initiative when conditions in the environment require so. Another property stated to distinguish agents from programs is temporal continuity [35], which lays the foundation for autonomy, social-ability, reactivity and pro-activeness.

Multi-agent systems is a subfield of distributed artificial intelligence which aims solving complex problems, that can not be solved by single agents, by using cooperating multiple agents. Multi-agent systems aid in the analysis and development of large-scale distributed systems and human-centered systems which are situated in complex, dynamic environments and require parallel processing [36]. That's why these systems are particularly recommended for developing systems composed of loosely coupled entities which can leave or join the system without any restrictions [37]. However, designing and programming multi-agent systems from scratch using plain programming languages is a hard task for engineers. This difficulty led to the proposal of various agent platforms and programming languages. In order to standardize these proposals Foundation of Intelligent Physical Agents of IEEE was formed in 1996. In 2005, FIPA standards were officially accepted.

Although there are many multi-agent systems, platforms and languages [38–40], only JADE, Jadex, JACK, EMERALD are fully FIPA compliant [39]. Among these platforms, JADE, Jadex and EMERALD are open source and free [41]. Since both Jadex and EMERALD are based on JADE [39], a JADE agent can run on both Jadex and EMERALD platforms [42,43]. This enables a prototyped system to be extended for Jadex system to use BDI agent architecture [43] or for EMERALD [42] system to use trust and reputation mechanisms. Among these three platforms, JADE is the most widely used one [44]. The reason for the widespread use of JADE is its ease to learn and develop system prototypes with [45–47] due to its active user community and many examples they have put on the Internet [39,48,49]. This ease of use is one of the reasons why JADE was chosen for prototyping the system introduced in this paper.

JADE is implemented using Java which makes it portable to different systems regardless of operating system or architecture. A JADE agent platform consists of multiple containers which may contain multiple agents. Since each JADE agent has a unique identifier, agents on different computers can communicate with each other transparently [44]. As a result of these features, a JADE platform may be distributed over multiple computers [39]. Additionally, since JADE is a scalable and fully FIPA compliant agent platform [44,50]. It provides the basic services like naming, yellow-pages (provided by the Directory Facilitator in FIPA compliant agent platforms), message-transport and parsing. JADE also has graphical debugging and deployment tools [41], a library of interaction protocols, an extendable kernel and supports the use of ontologies [39,44,50]. Scalability, FIPA compliance and portability of JADE are the other reasons for choosing it for prototyping our system.

2.2 Multi-agent Systems in E-Learning

E-learning systems, due to their inherent nature, are human-oriented, highly complex and distributed systems that require parallel processing [31,51–53]. Multi-agent systems are particularly recommended for modeling and implementing systems with these inherent properties. Therefore, in the literature there are many studies that utilize multi-agent technology for realizing e-learning systems [15–18,20–23,32,51–55]. There are also some recent studies that focus on multi-agent based e-learning system architectures developed with different agent development frameworks [16,22]. Programming environments of agents that contain e-learning materials and student models is another recent application area of multi-agent based e-learning systems [20,21].

There are three main types of agents in agent based e-learning systems: pedagogical, harvester and monitoring agents. Pedagogical agents are the most common type of agents found in e-learning systems [18,31,51–54]. These agents generally guide and motivate students during their studies and are represented as lifelike characters [56]. Pedagogical agents are generally interactive [52], and ask questions or provide solutions to achieve their goals [51]. Harvester agents are learning material collectors and generally run in parallel to harvest data and metadata from remote, heterogeneous repositories [19,22,32,51–53,55,57]. Finally, monitoring agents observe student activity within the system and provide a more personalized and enhanced learning experience by adapting the system via student profiling [15,16,18,54]. Some examples for what monitoring agents observe can be the types of learning materials preferred by the student, how much exercise is done while studying a subject, forum involvement [18], student performance during exercises and persistence on difficult tasks [54]. The educational components of agent based e-learning systems present in the literature can usually be classified based on these three agent types.

AgCAT catalog service [32] is a system that utilizes harvester agents only. There are three types of harvester agents in AgCAT: finder agents, librarian and inter-librarian agents. Finder agents search and retrieve learning objects registered by librarian agents. Librarian agents retrieve metadata from learning object repositories and store them in local catalogs. Inter-librarian agents federate learning object catalogs of remote AgCat systems.

Another adaptive system that utilizes agents is HAPA [53]. HAPA combines pedagogical and harvester agents. It is a system for supporting students in Java programming tasks too. Main focus of the system is code completion tasks. These tasks can be used as tests and exercises. HAPA harvester agents which run in parallel collect Java code examples from the web and give them to the classifier module which matches each code example with suitable lectures present in the system. Teachers using the system choose from these ready to use materials, select the parts to be completed by the students, and adds necessary hints to construct their lessons. These hints are used by the pedagogical agents. These agents interact with the student directly, observe his/her performance and adapt to him/her. They give the hints provided by the teacher and recommend additional learning material when student success decreases.

MagMAS [52] system is an example for adaptive systems that incorporate pedagogical agents which also act as monitoring agents. MagMAS is a system for Java programming tasks that focuses on code completion tasks. It is a re-implementation of the Mag system [58] by means of agents. MagMAS system uses pedagogical agents which also monitor the students [52]. In this system, marks and scores of each learner along with the time he/she spent on each material and past performances and learning history are monitored. The information gathered is used to build a learner model, which in turn is used by the pedagogical agents to recommend suitable learning materials, and give hints where necessary to each learner.

ElectronixTutor [54] is a system where pedagogical agents are used as tutors. This system aids in electronics education utilizing a recommender system. The system observe student behavior using a component called Learning Record Store which uses this data for building student models. Learning Record Store observes several components of the student performance like subject knowledge and grit while studying on topics. The resultant student model is used by the recommender system for material and tutor recommendation later on. There are specialized pedagogical agents for student profiles that can be classified as beginner, intermediate and advanced students. The recommender system decides which pedagogical agent to suggest depending on the student performance.

A system that utilizes pedagogical agents, which was also an inspiration for the system prototyped in this study, is eTeacher [18]. The pedagogical agent "eTeacher" in this system acts as a monitoring agent too. An overview of the "eTeacher" agent's functionality is shown in Fig. 1. The "eTeacher" agent observes students' interactions with the e-learning system inconspicuously. It gathers data about the student like his/her learning style, the number and type of the exercises done, his/her grades, and participation in chat rooms. A Bayesian network is used for finding out the students' learning styles. The student profile is then used by the "eTeacher" to provide guidance when needed regarding the materials to continue with, chats to participate in or questions to answer.

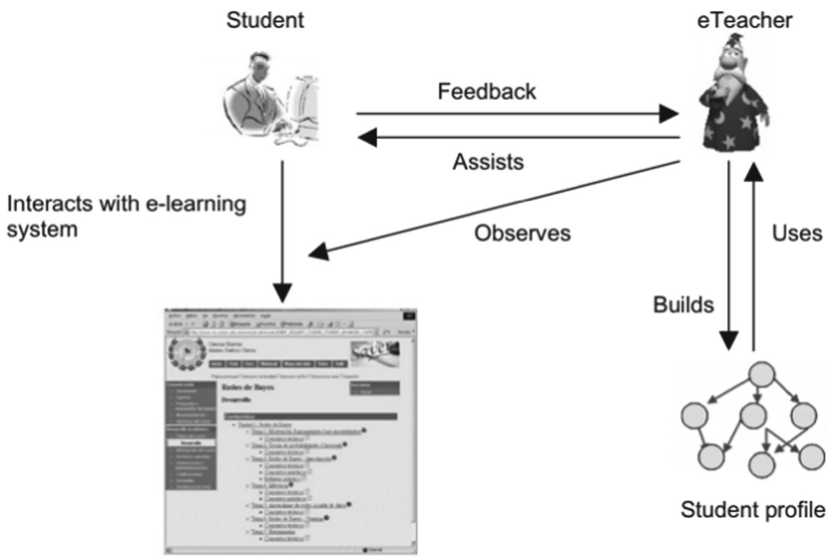

Fig. 1. An overview of the "eTeacher" agent's functionality in [18].

3 Multi-agent Based Adaptive E-Learning System

In this section, architecture of the proposed multi-agent based adaptive e-learning system is detailed based on related studies. After that, a prototype implementation of the proposed architecture is described.

3.1 Architecture of the Proposed E-Learning System

The architectural design of the proposed the multi-agent based adaptive e-learning system is based on related work such as [18,22,59].

As stated in the previous section, functionality that the agents offer in our system is inspired by the "eTeacher" agent in [18]. In a similar manner, we designed "student" agent type in our system that provides personalized learning materials to the students based on their learner models and keeps track of the interaction of the students with the e-learning system to assist them for an enhanced learning experience. Additionally, we designed another agent type called "lecturer" agent in our system that assists the lecturers in providing adaptive learning materials.

The proposed e-learning system is also based on the functionalities offered by intelligent tutoring systems. The architecture of the Protus intelligent tutoring system proposed by [59] is given in Fig. 2. Software agents were not used in [59]; however, the architecture provided in this study contains components such as domain module, learner model, application module, adaption module and session monitor that offer similar functionalities. The learner model in [59] provides similar functionality for building and using a student profile in the "eTeacher" system [18]. Besides, session monitor is also similar as it observes students' interactions with the system during a session. The "student" agent type in our system performs the operations supported by the adaption and application modules autonomously. Finally, the functionality of the domain module is provided by the "lecturer" agents in our system.

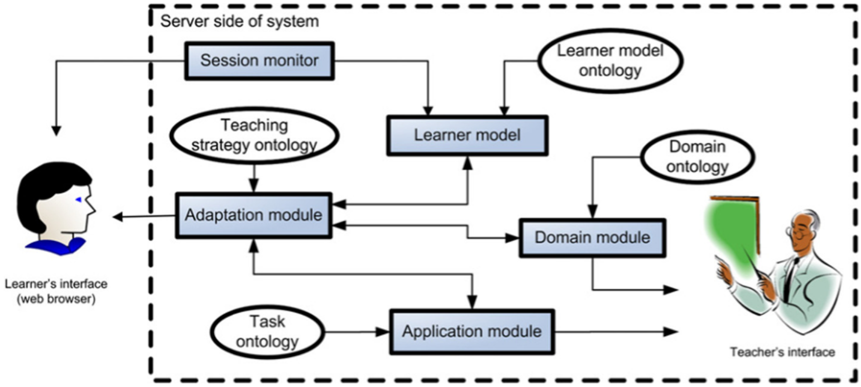

Fig. 2. The architecture of the Protus intelligent tutoring system in [59].

Another study that our system is based on is the multi-agent based architecture proposed by [22]. The architecture of [22] which is based on the Agents and Artifacts Metamodel [60], is demonstrated in Fig. 3. The environment modeling in multi-agent systems was emphasized in [22] and the focus was on the abstraction of the components in the environment of agents such as student models and learning object repositories with artifacts in a similar manner to [20,21]. In this study, we focused fully on how to support the functionalities provided by the agents in an intelligent way, and therefore, decided to skip the environment modeling aspect of multi-agent system programming.

As a result of the literature review, we decided to design an architecture that focuses on functionalities supported by the related studies. Thus, we designed two types of agents named as "student" and "lecturer" and focused on the interaction of them. As the names suggest, these agents represent and are used by students and lecturers, respectively.

"Student" agents monitor the interaction between the student and the system. The behaviors observed include the time spent on certain types of learning materials and mistakes made during learning sessions. "Student" agents are also responsible for choosing the learning materials suitable for the student. As students engage with these materials, further observations are done, gathered data

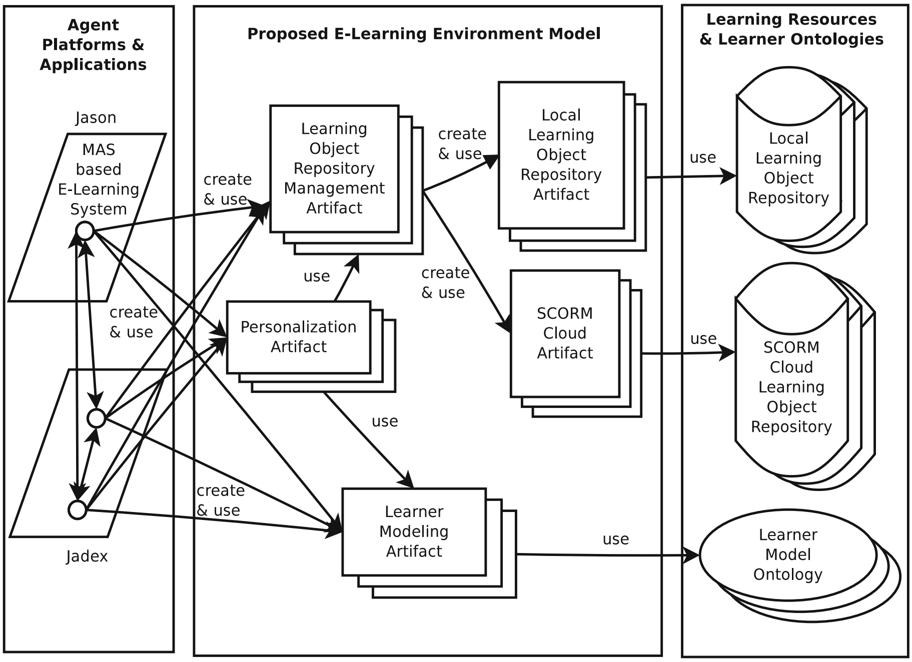

Fig. 3. The adaptive e-learning environment architecture based on Agents and Artifacts Metamodel proposed in [22].

are processed and the resulting data are stored in the student model. If success level of a student an agent is responsible for drops significantly, a notification is sent to the "lecturer" agent for further reasoning about the student.

"Lecturer" agents request metadata from lecturers for the learning materials they supply. "Lecturer" agents work collaboratively with "student" agents, process the student data on notifications they received from "student" agents, and assist the students by means of the information they supply to student agents if they are at risk of dropping the course. When the activities of both agent types are considered, both of them can be considered as pedagogical and monitoring agents.

The architecture of the proposed adaptive e-learning system is demonstrated in Fig. 4. In our architecture, "student" and "lecturer" agents communicate with their users via separate interfaces. All of the agents register themselves to the DF (Directory Facilitator) agent. This enables them to search for other agents via DF and communicate with them using agent messages later on. Student modeling layer includes student models in which student information such as demographic information, learning styles, courses taken and interaction history with the e-learning system are stored. Learning materials layer contains learning contents personalized according to learning styles.

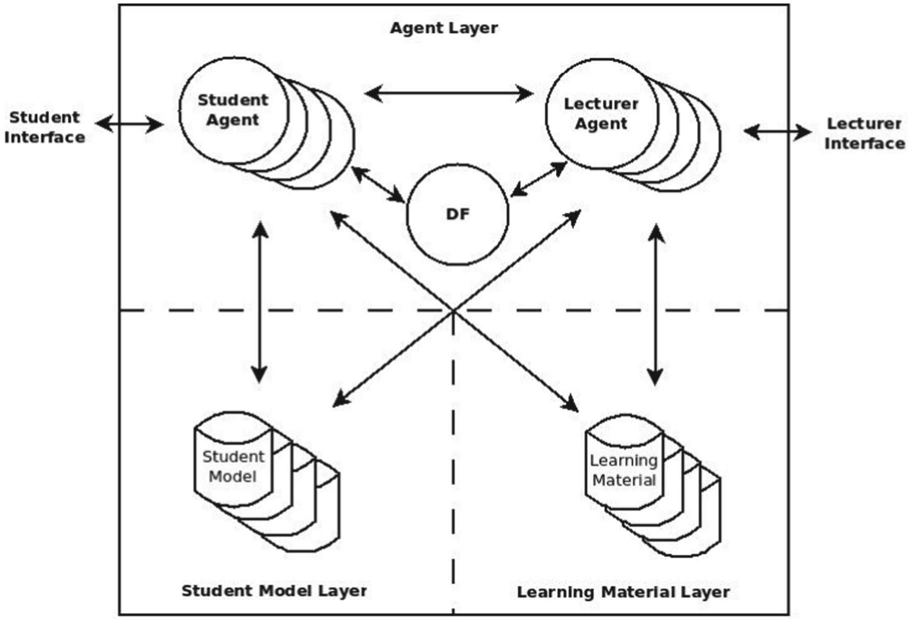

Fig. 4. Architecture of the proposed multi-agent based adaptive e-learning system.

3.2 Prototype Implementation

We implemented a prototype of the proposed adaptive e-learning architecture with JADE agent development platform. JADE is one of the most commonly used agent platforms which provides a GUI based interface. There are e-learning systems such as [16,19] that have been successfully implemented with JADE. Thus, we decided to use JADE for the prototype implementation.

Ontologies serve as a common vocabulary between software agents and enhance system interoperability. This is also true for e-learning systems [61]. In addition to this, ontologies facilitate learning material description, sharing and search on the Web [61]. There are many studies that use ontologies for student and learning material modeling [16,23,27,28,30,59]. Thus, we decided to use ontological representation for student modeling in our prototype. An ontological student model based on Felder and Silverman, Kolb and Honey-Mumford learning style models was used in [23]. In our prototype, we utilized a subset of this model [23] and focused only on the Felder and Silverman learning style model. However, we extended this model subset to store the interaction history of students with the e-learning system.

JADE provides the Remote Agent Management GUI shown in Fig. 5 where agents running on the prototype can be observed and managed. A student agent named “learner1” and a lecturer agent called “provider1” running on JADE can be seen in Fig. 5.

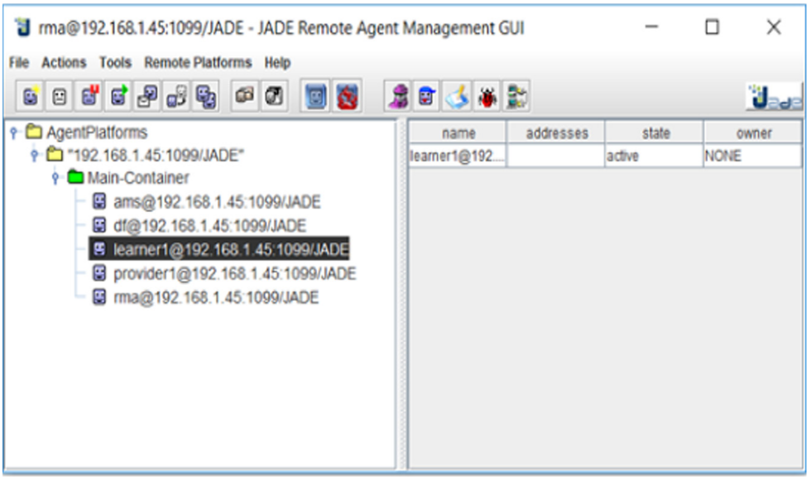

Fig. 5. Remote Agent Management GUI of the prototype in JADE agent platform.

An example screenshot of a GUI where lecturers add a learning material to the system is shown in Fig. 6. In this GUI, the lecturer enters metadata of a learning material such as title, description, keywords and learning style, and then, selects the source file of the learning material.

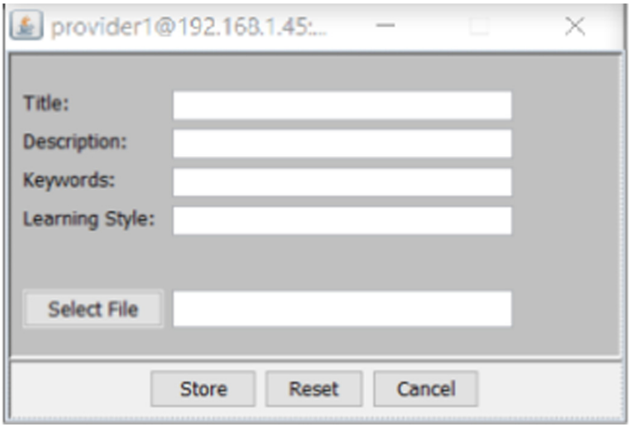

Fig. 6. An example GUI of the prototype for adding learning materials.

The functions provided at this stage of the prototype implementation are adding learning materials to the system, learning material recommendation according to student learning style (which is implemented based on our previous work in [62]), and determining at risk students who interact with the system at a lower rate than the average interaction rate. These functions are still under development and will be enhanced as the implementation goes on.

4 Conclusion

In this paper, we presented a multi-agent based adaptive e-learning system that is prototyped with JADE agent platform. By providing adaptivity based on learning styles of the students, it is aimed to provide higher success levels compared to non-personalized e-learning systems. The multi-agent based nature of the system provides an effective control mechanism to monitor the interaction of students with the system and to observe their learning levels. We used an ontology to model students to increase re-usability of the student model and interoperability of the proposed system.

As future work, a learning material ontology can be developed to provide a modular ontological model that can be used by other e-learning systems, too. In this regard, the prototype implementation with JADE can be extended to utilize other agent platforms such as Jason and Jadex. Additionally, in the following phases of the prototype implementation, we plan to add enhanced reasoning capability to the lecturer agent.

This study focuses on a multi-agent based architecture proposal. Implementation of the prototype still continues. When the prototype is fully implemented, we plan to test the impact of the architecture on student performance. In order to conduct these tests, personalized learning materials should be prepared. In this context, as another future work, it is planned to use the proposed system in a course where the educational content is personalized with Felder and Silverman learning styles model.

Acknowledgments. This study was supported by Ege University Scientific Research Projects Coordination Unit (Project number 18-MUH-035).

References

1. Soni, V.D.: Global Impact of E-learning during COVID 19 (2020). Available at SSRN: https://ssrn.com/abstract=3630073
2. Almaiah, M.A., Al-Khasawneh, A., Althunibat, A.: Exploring the critical challenges and factors influencing the E-learning system usage during COVID-19 pandemic. Educ. Inf. Technol. **25**(6), 5261–5280 (2020). https://doi.org/10.1007/s10639-020-10219-y
3. Elumalai, K.V., et al.: Factors affecting the quality of E-learning during the COVID-19 pandemic from the perspective of higher education students. In: COVID-19 and Education: Learning and Teaching in a Pandemic-Constrained Environment, p. 189 (2021)
4. Sangineto, E., Capuano, N., Gaeta, M., Micarelli, A.: Adaptive course generation through learning styles representation. Univ. Access Inf. Soc. **7**(1–2), 1–23 (2008). https://doi.org/10.1007/s10209-007-0101-0

5. Essalmi, F., Ayed, L.J.B., Jemni, M., Kinshuk, Graf, S.: A fully personalization strategy of e-learning scenarios. Comput. Hum. Behavior **26**(4), 581–591 (2010)
6. Ciloglugil, B., Inceoglu, M.M.: User modeling for adaptive e-learning systems. In: Murgante, B., et al. (eds.) ICCSA 2012. LNCS, vol. 7335, pp. 550–561. Springer, Heidelberg (2012)
7. Akbulut, Y., Cardak, C.S.: Adaptive educational hypermedia accommodating learning styles: a content analysis of publications from 2000 to 2011. Comput. Educ. **58**(2), 835–842 (2012)
8. Truong, H.M.: Integrating learning styles and adaptive e-learning system: current developments, problems and opportunities. Comput. Hum. Behav. **55**, 1185–1193 (2015)
9. Ozyurt, O., Ozyurt, H.: Learning style based individualized adaptive e-learning environments: content analysis of the articles published from 2005 to 2014. Comput. Hum. Behav. **52**, 349–358 (2015)
10. Ciloglugil, B.: Adaptivity based on Felder-Silverman learning styles model in E-learning systems. In: 4th International Symposium on Innovative Technologies in Engineering and Science, ISITES 2016, pp. 1523–1532 (2016)
11. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. Eng. Educ. **78**(7), 674–681 (1988)
12. Gregg, D.G.: E-learning agents. Learn. Organ. **14**, 300–312 (2007)
13. Khemakhem, F., Ellouzi, H., Ltifi, H., Ayed, M.B.: Agent-based intelligent decision support systems: a systematic review. IEEE Trans. Cogn. Dev. Syst., 1 (2020). https://doi.org/10.1109/TCDS.2020.3030571
14. Ciloglugil, B., Inceoglu, M.M.: Developing adaptive and personalized distributed learning systems with semantic web supported multi agent technology. In: 10th IEEE International Conference on Advanced Learning Technologies, ICALT 2010, Sousse, Tunesia, 5–7 July 2010, pp. 699–700. IEEE Computer Society (2010)
15. Dung, P.Q., Florea, A.M.: An architecture and a domain ontology for personalized multi-agent e-learning systems. In: Third International Conference on Knowledge and Systems Engineering, KSE 2011, pp. 181–185. IEEE. (2011)
16. Rani, M., Nayak, R., Vyas, O.P.: An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage. Knowl.-Based Syst. **90**, 33–48 (2015)
17. Sun, S., Joy, M., Griffiths, N.: The use of learning objects and learning styles in a multi-agent education system. J. Interact. Learn. Res. **18**(3), 381–398 (2007)
18. Schiaffino, S., Garcia, P., Amandi, A.: eTeacher: providing personalized assistance to e-learning students. Comput. Educ. **51**(4), 1744–1754 (2008)
19. Sandita, A.V., Popirlan, C.I.: Developing a multi-agent system in JADE for Information management in educational competence domains. Procedia Econ. Finance **23**, 478–486 (2015)
20. Ciloglugil, B., Inceoglu, M.M.: Exploiting agents and artifacts metamodel to provide abstraction of E-learning resources. In: 17th IEEE International Conference on Advanced Learning Technologies, ICALT 2017, Timisoara, Romania, 3–7 July 2017 (2017)
21. Ciloglugil, B., Inceoglu, M.M.: An agents and artifacts metamodel based E-learning model to search learning resources. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10404, pp. 553–565. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62392-4_40
22. Ciloglugil, B., Inceoglu, M.M.: An adaptive E-learning environment architecture based on agents and artifacts metamodel. In: 18th IEEE International Conference on Advanced Learning Technologies, ICALT 2018, Mumbai, India, 9–13 July 2018 (2018)

23. Ciloglugil, B., Inceoglu, M.M.: A learner ontology based on learning style models for adaptive E-learning. In: Gervasi, O., et al. (eds.) ICCSA 2018. LNCS, vol. 10961, pp. 199–212. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95165-2_14
24. Marik, V., Gorodetsky, V., Skobelev, P.: Multi-agent technology for industrial applications: barriers and trends. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1980–1987 (2020)
25. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. **284**(5), 34–43 (2001)
26. Ciloglugil, B., Inceoglu, M.M.: Ontology usage in E-learning systems focusing on metadata modeling of learning objects. In: International Conference on New Trends in Education, ICNTE 2016, pp. 80–96 (2016)
27. Essalmi, F., Ayed, L.J.B., Jemni, M., Kinshuk, Graf, S.: Selection of appropriate e-learning personalization strategies from ontological perspectives. Interact. Des. Archit. J. IxD&A **9**(10), 65–84 (2010)
28. Valaski, J., Malucelli, A., Reinehr, S.: Recommending learning materials according to ontology-based learning styles. In: Proceedings of the 7th International Conference on Information Technology and Applications, ICITA 2011, pp. 71–75 (2011)
29. Yarandi, M., Jahankhani, H., Tawil, A.R.H.: A personalized adaptive e-learning approach based on semantic web technology. Webology **10**(2), Art-110 (2013)
30. Kurilovas, E., Kubilinskiene, S., Dagiene, V.: Web 3.0-based personalisation of learning objects in virtual learning environments. Comput. Hum. Behav. **30**, 654–662 (2014)
31. Gago, I.S.B., Werneck, V.M.B., Costa, R.M.: Modeling an educational multi-agent system in MaSE. In: Liu, J., Wu, J., Yao, Y., Nishida, T. (eds.) AMT 2009. LNCS, vol. 5820, pp. 335–346. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04875-3_36
32. Barcelos, C., Gluz, J., Vicari, R.: An agent-based federated learning object search service. Interdisc. J. E-Learn. Learn. Objects **7**(1), 37–54 (2011)
33. Norvig, P.R., Intelligence, S.A.: A Modern Approach. Prentice Hall, Upper Saddle River (2002)
34. Wooldridge, M.J., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**(2), 115–152 (1995)
35. Franklin, S., Graesser, A.: Is It an agent, or just a program?: a taxonomy for autonomous agents. In: Müller, J.P., Wooldridge, M.J., Jennings, N.R. (eds.) ATAL 1996. LNCS, vol. 1193, pp. 21–35. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0013570
36. Julian, V., Botti, V.: Multi-agent systems. Appl. Sci. **9**(7), 1402 (2019). MDPI AG
37. Mariani, S., Omicini, A.: Special issue "multi-agent systems". Appl. Sci. **9**(5), 954 (2019). MDPI AG
38. Leon, F., Paprzycki, M., Ganzha, M.: A review of agent platforms. In: Multi-Paradigm Modelling for Cyber-Physical Systems (MPM4CPS), ICT COST Action IC1404, pp. 1–15 (2015)
39. Kravari, K., Bassiliades, N.: A survey of agent platforms. J. Artif. Soc. Soc. Simul. **18**(1), 11 (2015)
40. Bordini, R.H., et al.: A survey of programming languages and platforms for multi-agent systems. Informatica **30**(1), 33–44 (2006)
41. Pal, C. V., Leon, F., Paprzycki, M., Ganzha, M.: A review of platforms for the development of agent systems. arXiv preprint arXiv:2007.08961 (2020)

42. Kravari, K., Kontopoulos, E., Bassiliades, N.: EMERALD: a multi-agent system for knowledge-based reasoning interoperability in the semantic web. In: Konstantopoulos, S., Perantonis, S., Karkaletsis, V., Spyropoulos, C.D., Vouros, G. (eds.) SETN 2010. LNCS (LNAI), vol. 6040, pp. 173–182. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12842-4_21
43. Braubach, L., Lamersdorf, W., Pokahr, A.: Jadex: implementing a BDI-infrastructure for JADE agents. EXP Search Innov. **3**(3), 76–85 (2003)
44. Bergenti, F., Caire, G., Monica, S., Poggi, A.: The first twenty years of agent-based software development with JADE. Auton. Agents Multi-Agent Syst. **34**(2), 1–19 (2020). https://doi.org/10.1007/s10458-020-09460-z
45. Balachandran, B.M., Enkhsaikhan, M.: Developing multi-agent E-commerce applications with JADE. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007. LNCS (LNAI), vol. 4694, pp. 941–949. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74829-8_115
46. Zhao, Z., Belloum, A., De Laat, C., Adriaans, P., Hertzberger, B.: Using Jade agent framework to prototype an e-Science workflow bus. In: Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007), pp. 655–660 (2007)
47. Kularbphettong, K., Clayton, G., Meesad, P.: e-Wedding based on multi-agent system. In: Demazeau, Y., et al. (eds.) Trends in Practical Applications of Agents and Multiagent Systems. Advances in Intelligent and Soft Computing, vol. 71, pp. 285–293. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12433-4_34
48. van Moergestel, L., Puik, E., Telgen, D., van Rijn, R., Segerius, B., Meyer, J.J.: A multiagent-based agile work distribution system. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 2, pp. 224–230. IEEE (2013)
49. Scutelnicu, A., Lin, F., Kinshuk, Liu, T., Graf, S., McGreal, R.: Integrating JADE agents into moodle. In: Proceedings of the International Workshop on Intelligent and Adaptive Web-Based Educational Systems, pp. 215–220 (2007)
50. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-agent Systems with JADE, vol. 7. Wiley, Hoboken (2007)
51. Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z., Jain, L.C.: Agents in E-learning environments. In: E-Learning Systems. ISRL, vol. 112, pp. 43–49. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-41163-7_5
52. Ivanović, M., Mitrović, D., Budimac, Z., Vesin, B., Jerinić, L.: Different roles of agents in personalized programming learning environment. In: Chiu, D.K.W., Wang, M., Popescu, E., Li, Q., Lau, R. (eds.) ICWL 2012. LNCS, vol. 7697, pp. 161–170. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43454-3_17
53. Ivanović, M., Mitrović, D., Budimac, Z., Jerinić, L., Bădică, C.: HAPA: harvester and pedagogical agents in e-learning environments. Int. J. Comput. Commun. Control **10**(2), 200–210 (2015)
54. Graesser, A.C., et al.: ElectronixTutor: an intelligent tutoring system with multiple learning resources for electronics. Int. J. STEM Educ. **5**(1), 15 (2018)
55. Sharma, S., Gupta, J.P.: A novel architecture of agent based crawling for OAI resources. Int. J. Comput. Sci. Eng. **2**(4), 1190–1195 (2010)
56. Heidig, S., Clarebout, G.: Do pedagogical agents make a difference to student motivation and learning? Educ. Res. Rev. **6**(1), 27–54 (2011)

57. De la Prieta, F., Gil, A.B.: A multi-agent system that searches for learning objects in heterogeneous repositories. In: Demazeau, Y., et al. (eds.) Trends in Practical Applications of Agents and Multiagent Systems. Advances in Intelligent and Soft Computing, vol. 71, pp. 355–362. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12433-4_42
58. Ivanović, M., Pribela, I., Vesin, B., Budimac, Z.: Multifunctional environment for e-learning purposes. Novi Sad J. Math. **38**(2), 153–170 (2008)
59. Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z.: E-Learning personalization based on hybrid recommendation strategy and learning style identification. Comput. Educ. **56**(3), 885–899 (2011)
60. Ricci, A., Piunti, M., Viroli, M.: Environment programming in multi-agent systems: an artifact-based perspective. Auton. Agent. Multi-Agent Syst. **23**(2), 158–192 (2011). https://doi.org/10.1007/s10458-010-9140-7
61. Dalipi, F., Idrizi, F., Rufati, E., Asani, F.: On integration of ontologies into e-learning systems. In: 2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks, pp. 149–152. IEEE (2014)
62. Ciloglugil, B., Inceoglu, M.M.: A Felder and Silverman learning styles model based personalization approach to recommend learning objects. In: Gervasi, O., et al. (eds.) ICCSA 2016. LNCS, vol. 9790, pp. 386–397. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42092-9_30