Indonesian Journal of Electrical Engineering and Computer Science
Vol. 22, No. 3, June 2021, pp. 1679~1687
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v22.i3.pp1679-1687

Multi-agent class timetabling for higher educational institutions using prometheus platform

Angelita D. Guia[1], Melvin A. Ballera[2]
[1]University of the East, Caloocan Campus, Philippines
[2]Research and Development Office, Technological Institute of the Philippines, Manila Philippines

Article Info

Article history:

Received Jan 14, 2021
Revised Jun 12, 2021
Accepted Jun 13, 2021

Keywords:

Class timetabling
Higher educational institutions
Multi-agent system
Prometheus methodology

ABSTRACT

A in a university setting, class scheduling is vital for teaching and learning process. Academic institutions rely on time tables in their day to day activities. University Course Timeframe problem can be resolved by using multi-agent systems-based method which may increase the independence of each department's class scheduling, adaptability in a distributed environment and prevents conflicts between events or resources, and unforeseen allocation through intervention between agents in a dispersed environment. Class timing is performed manually in most of the higher educational institutions, which is a very challenging and time-consuming process. The main objective of the study is to build a multi-agent class timing system that automates the process of class scheduling of higher education institutions (HEIs) using the Prometheus methodology. The implementation of the Prometheus approach in the development of a multi-agent framework has resulted in a complete and comprehensive system covering all phases of software development as applied to the agent systems.

Corresponding Author:

Angelita D. Guia
College of Engineering
University of the East
105 Samson Road, Caloocan City 1400, Metro Manila, Philippines
Email: angelita.guia@ue.edu.ph

1. INTRODUCTION

Currently the interplay between innovation and digitalization are among the most important assets for educational system. Higher educational institutions concentrate on discovering new knowledge by means of digitalization methods and endorse new forms of its distribution in the society [1]. New opportunities for academic institutions are enabled by adopting digital technologies and incorporating it in their infrastructure [2], one of which is the automation of its class timetabling system. Class preparation is important in a university setting in the teaching and learning process. The educational institutions use timetables for their everyday operations. Timetable management means planning the use of scarce resources as ideally as possible, promising effective teaching and learning and safeguarding the comfort of stakeholders as much as possible. Timetabling, which is a time-consuming job, is designed for any semester or term. Multi-agent systems can be used to answer problems that are hard or difficult for each agent or a monolithic system to solve and it became a widespread approach in solving problems [3]. Multi-agent approaches can be used to solve school timetabling problems. These include an improvement in the independence of the establishment of schedules for each agency, adaptability in a disseminated setting, and the prevention of conflict between events/resources, and unintended distribution by cooperation between agents in a distributed environment [4]. A multi-agent framework is made up of software agents that work in unity to solve problems beyond the abilities of each agent. The common objective of the different agent is to enhance the value of class timetabling solutions until the appropriate solution is establish when the end condition was satisfied [5].

Agents are tools that are inspired by the global environment in order to build initial systems instances. Each time a multi-agent framework is considered, it can be said that there is a group of agents interacting together to resolve problems that are beyond the abilities of each agent. An agent can detect and identify something from the surrounding sensors and then run through the driver through the atmosphere [6]. Agents are characterized into various categories based on their presentation, which includes: i) independent, ii) smart, iii) sensitive, iv) constructive, v) novice, vi) itinerant, and vii) communicative agents.

The purpose of the university course timeframe program (UCTTP) is to search for a technique for assigning all activities to the predefined timeframes and rooms where all constraints must be met. Class timetable activities include pupils, instructors, and classes, where services include room facilities and facilities such as lecture rooms and lab rooms. Timeframes also involve two primary segments, namely daily and weekly timeframes, which vary from school to school. However, every room in a school contains different tools, including audio visual resources, quantity of chairs per classroom, number of boards relevant to each lecture and laboratory room and others. [4] Itinerant multi-agent system scheme can deliver a practical and versatile method of designing a timeline that solves the problem in a fragmented and active way [7], [8].

Using multi-agent systems technique, the time tabling problem has been divided into a set of simpler problems, each of them is handled by a separate agent. It is very easy to handle very complicated situations by augmenting the agent with the knowledge and rules to handle these situations. Another benefit of using multi-agents is reusability. Due to the inherent modularity of Multi-Agent Systems, it is easy to use the same agents in solving problems related to resource allocation and management problems, in general [9].

However, in higher institutions, there are no intelligent time table systems and class scheduling is designed manually in many universities [10]. The task is hard and laborious [11]. Other problems of a traditional manual system include lack of accuracy, slow speed, and low information sharing. The goal of class timetabling is to discover an applicable timetable for the different classes to be scheduled with limited resources, such as rooms and class schedules [12]. The class schedule must be created while concurrently satisfying different constraints like effective use of resources, satisfaction of the students and instructors and others [13].

In this study, a class scheduling system is developed using multi-agent-based architecture to automate the class scheduling system of the higher educational institutions (HEIs) and to solve the problems encountered in the manual system. Figure 1 illustrates the conceptual framework of the study. Multi-agent system is used in developing the software. An automated class schedule for each year level will be is generated based on the curriculum, resources, and teacher schedule and specialization.

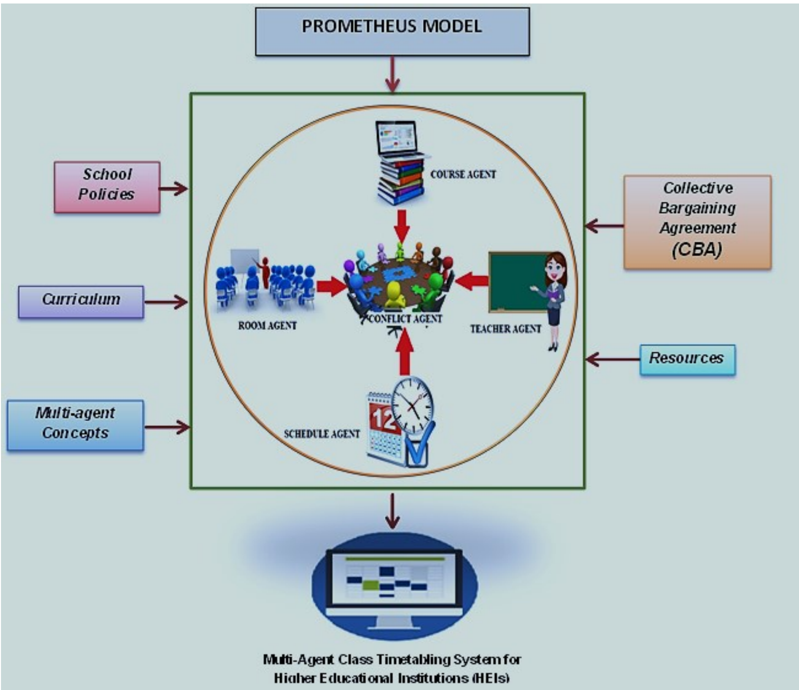

Figure 1. Conceptual framework

The main objective of the study is to determine if the scheduling of classes in higher educational institutions can be automated by developing a multiple agent class timetabling system using Prometheus platform. The scheduling designed manually and the scheduling done through multi-agent class timetabling system are then compared to conclude if there is a significant difference between the two and if the automation of schedules is effective. Specifically, the study aims to: i) Identify the agents and constraints; ii) Optimize computing concepts with illustrative examples; iii) Determine the sequence and activities of different agents; iv) Determine the effectiveness of the multi-agent system; and v) Determine if there is a significant difference between manual and automated class timetable.

2. RESEARCH METHOD

The method of product creation is analyzed and defined and the final product is evaluated [14]. In the software development environment, multi-agent systems are gaining interest. The fast growth of the development of multi-agent systems relies on the idea that the agent model is ideal for exploring the opportunities provided by open distributed systems such as internet [15]. One of the most explored features of multi-agent systems is communication and interaction. In order to reflect interaction among agents, all the methodologies studied use special models. Agent internal representation, on the contrary, is ignored by most methodologies. Only Prometheus attempts to catch the internal agents [16].

This research utilizes a multi-agent timetabling method using the Prometheus Platform. It is a systematic technique of defining, planning, applying and debugging agent oriented systems. Figure 2 shows the three phases of the process, namely: i) the system specification phase, which concentrates on the identification of the objectives and the basic intent of the system, together with the inputs and outputs; ii) the architectural design phase, which relies on the actions of the preceding phase; and iii) the comprehensive design phase, which concerns the internal actions of each agent and how the agent performs. [17]. Constraints in class schedule problem are classified into two: hard and soft constraints [18]-[22]:

a. Hard constraints must be fulfilled totally so that the created solutions are conceivable and without conflict. This must not create any violation. The following are the hard constraints:
   - There is only one course presentation session for each particular course every day.
   - The features of class are to be considered over the presented course.
   - Each pair of student and teacher is present only at the same time in one class.
   - One course must be assigned in one timeslot and one classroom.
   - Teachers must be available at the timeslots and courses scheduled for them.
   - Certain courses need constant and specified number of sessions at every week (for example, 3, 2, 1 times per week).
   - Each course must be scheduled at a particular time (following the priority).
b. Soft constraints, which are related to objective function, must be fulfilled as much as possible. However, they are not necessarily satisfied as hard constraints are. The following are the soft constraints:
   - The teacher considers a time priority to teach.
   - Empty spaces must be eliminated within classrooms.
   - The maximum continuous teaching time for a teacher must be three hours.
   - The maximum continuous class time for students is four hours.
   - A subject must not be taught for more than two consecutive hours.
   - Uniform distribution of courses among timeslots should be tried.
   - Teachers and students should have lunch hours between 12 p.m. to 1 p.m. or 1 p.m. to 2 p.m.
   - Students must not have only one timeslot (for one hour) for one class in a day.

To solve these constraints, the following agents are developed (Figure 2):

- **Course agent** is responsible for the course information. Curriculum template for each course is added. This curriculum template is the basis of course offering for each year level per semester of each course in the university.
- **Teacher agent** is responsible for the teacher information like status (full-time or part-time), specialization and time preference. With these details, basis for scheduling the subjects and solving constraints can be established.
- **Schedule agent** is responsible for the available timeslot for each course. This handles the identification of allotted time per subject.
- **Room agent** is responsible for the available room for each course and each timeslot. Type of classroom (eg. lecture or laboratory room), room capacity is also handled by this agent. Available hardware resources (eg. Projector and equipment) and software resources (eg. Visual Basic, AutoCAD, and Matlab) per room are also added.

– **Conflict agent** is responsible for proper communication or negotiation with all the agents within the system. This is the heart of the system because this handles all the conflicts among the agents.

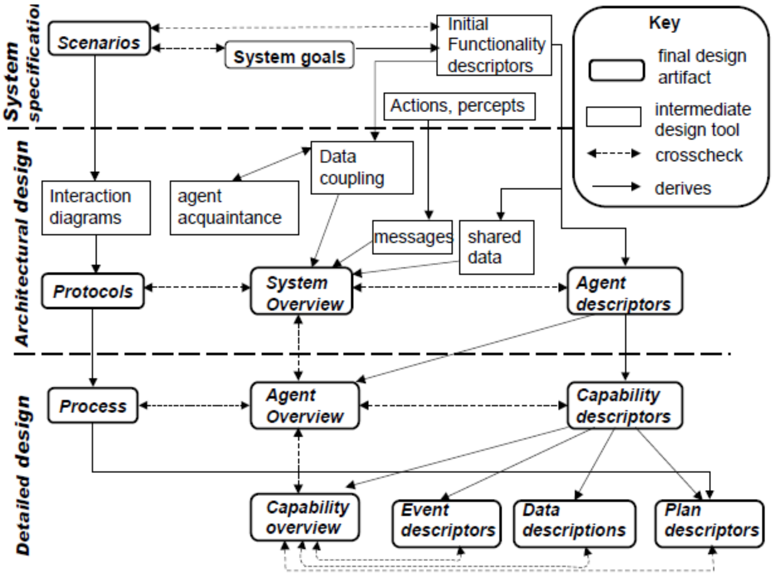

Figure 2. The phases of Prometheus methodology

3. RESULTS AND DISCUSSION

Figure 3 shows the solutions to the specific research problems. The system comprises of five agents, namely: course agent, room agent, teachers agent, schedule agent and conflict agent. Actual course timetabling in the university environment focuses on maintaining a fair allocation of courses the timetable with minimal conflicts specifically for regular students to complete their educational program by terms, semesters, and years. The solution must efficiently map the inherent and intrinsic features and existence of agents and agent systems in order to create an effective and powerful agent solution to any problems [23]. Therefore, in this research above principle is observed by mapping agent autonomy into five agents that comprises the system and is communicating with each other in order to produce an efficient timetable for classes in the HEIs. The study [13] concentrates on agents that are designed based on the belief-desire-intention (BDI) Architecture as shown in Figure 4. The BDI architecture is adopted in the design of the different agents.

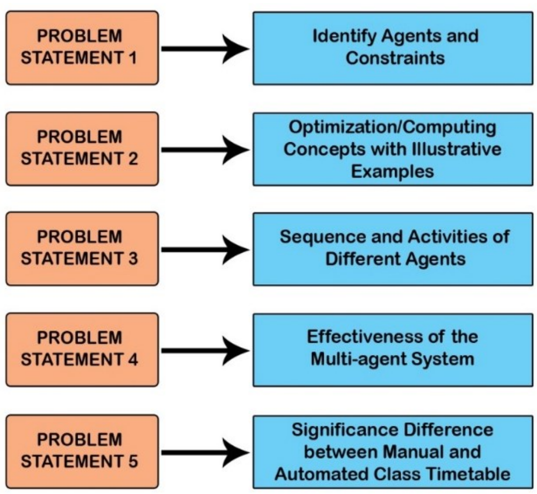

Figure 3. Solutions to research problems

```
B ← B0  /* B0 are initial beliefs*/
I ← I0  /* I0 are initial intentions*/
while true do
     get next percept ρ via sensors
    B ← brf(B, ρ)
    D ← options( B,I)
    I ← filter(B,D,I)
    H ← plan(B,I,Ac)   /* Ac is the set of actions*/
    While not (empty(H) or succeeded(I,B) or impossible (I,B) do
          α ← first element of H;
          execute (α);
          H ← tail of H;
          observe environment to get next percept ρ;
          B ← brf(B, ρ)
          if reconsider(I,B) then
               D ← options( B,I)
               I ← filter(B,D,I)
               end-if
               if not sound(H,I,B) then
                  H ← plan(B,I,Ac);
               end-if
          end-while
end-while
```

Figure 4. A Computational model of belief-desire-intention architecture

For the study, Python is used as a tool for developing the system. Python has proven to be a suitable programming language for implementing a multi-agent system. The multi-agent class timetabling system for higher educational institutions system should be able to do the following things as shown in Figure 5: i) Allow for input of teacher data; ii) Allow for input of courses data; iii) Allow for input of room data; iv) Allow for input of resources data; v) Allow to automatically generate class schedules per section per course; and vi) Allow for notification if there are errors in the creation of class schedules.

A clear understanding of the system must be developed in this phase. How the agent reacts (action) to the inputs from the environment (percept) have to be identified. The goals and sub-goals, appropriate scenarios and roles have to be identified also. A worthy scheduling method that can lead to optimization is significant to make sure it is able to create the needed class timetable [24]. With this, a mathematical model is needed for the optimization process. The study of Sen describes a mathematical model for the class scheduling since it is a combinatorial problem. Solving a course scheduling problem is equivalent to finding a subset and satisfying the constraints [25].

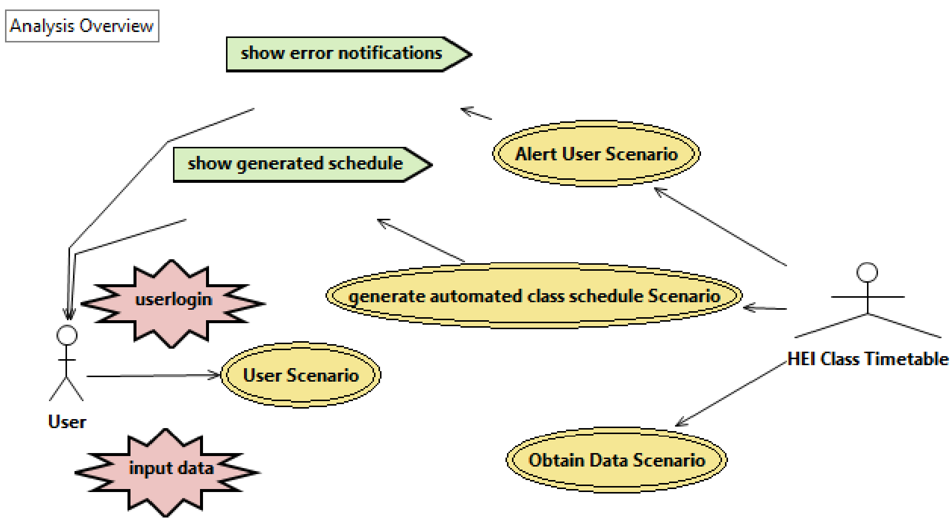

Figure 5. Analysis overview of the system

Assuming that there is a set of ten classes and five faculty members, that all the faculty members can teach all the courses, and that there are three rooms and five timeslots for each day, the following sets can be derived:
- Classes, C = {C1, C2, C3, C4, C5, C6, C7, C8, C9, C10}
- Time Slots, T = {T1, T2, T3, T4, T5}
- Rooms, R = {R1, R2, R3}
- Faculty members, F = {F1, F2, F3, F4, F5}
- A solution for this will be a set:

Schedule, S = {S1, S2, S3, S4, … }

The Timetable formula is:

$$Timetablingi = (C \times T \times R \times F) \quad (1)$$

where C is the set of classes, T is the set of class time slots, R is the set of rooms, and F is the set of faculty members. Then $Timeslot_i$ is a high-dimensional and multi-constraint combinatorial optimization problem. Where, each $Timeslot_i$ is a four-tuple contained in C×R×T×F and satisfying the constraints. Since the mathematical model is ready, the process continues and introduce a few constraints into the system can be introduced. This mathematical model was adopted in the study. Each sub-problem is tackled individually and then fed into the other sub-problem as shown in Figure 6. By introducing modularity in the solution different algorithms to solve each sub-problem can now be employed. Then the output of each sub-solution can be fed as the input to the other for improved results. For this research, the scope has been limited to have similar algorithms at each node: i) Mapping of the courses based of the teachers' specialization and time availability; ii) Mapping of the courses to rooms; and iii) Assigning a timeslot to the lecture.

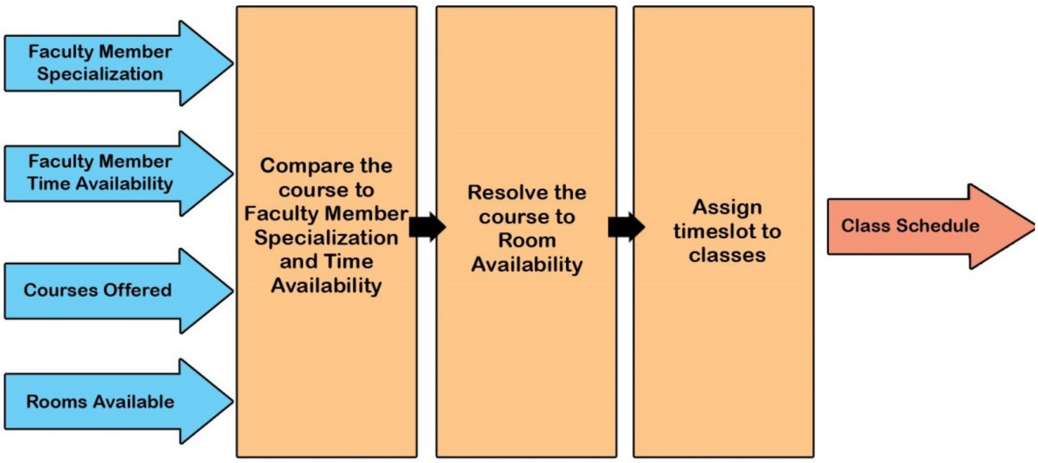

Figure 6. Course scheduling problem (CSP) solution model

The solution to the course scheduling problem (CSP) consists of the following main sub-problems as depicted in Figure 6. To evaluate the system, a class schedule created manually is matched to the class schedule generated manually by the system in terms of the subject load for teachers. Table 1 presents the difference in the number of units per teacher in the department using the manual system against the automated class scheduling system. The difference between the manual and automated scheduling of the system in terms of the number of units per faculty is also considered. The teaching load (courses and schedules) of the nine teachers of the Computer Engineering Department are used and are compared with the number of units produced by the automated scheduling.

Table 2 shows the result of the test using independent sample t-test. The mean for manual scheduling is 15.11 and that of automated scheduling is also 15.11 mean with a standard deviation 8.038. The test results indicate that there is no significant difference between the two means, with p value=0.865, which is much greater than 0.05 or 0.01 level of significance. Considering the test results, the null hypothesis is accepted: there is no significant difference between the manual scheduling and the automated scheduling. This means that the scheduling of classes can be automated.

Table 1. Comparison between manual and automated class schedule

Class Schedule	Teachers	Manual system	Automated system	Difference
1ST SEM SY 2018-2019	Teacher 1	21	20	1 (5% ↑)
	Teacher 2	23	20	3 (15% ↑)
	Teacher 3	7	7	0 (0%)
	Teacher 4	13	15	-2 (13.33% ↓)
	Teacher 5	2	2	0 (0%)
	Teacher 6	14	17	-3 (17.65% ↓)
	Teacher 7	24	24	0 (0%)
	Teacher 8	24	24	0 (0%)
	Teacher 9	10	10	0 (0%)
2ND SEM SY 2017-2018	Teacher 1	11	11	0 (0%)
	Teacher 2	4	7	-3 (42.86% ↓)
	Teacher 3	6	6	0 (0%)
	Teacher 4	11	11	0 (0%)
	Teacher 5	3	3	0 (0%)
	Teacher 6	6	6	0 (0%)
	Teacher 7	24	21	3 (14.29% ↑)
	Teacher 8	7	7	0 (0%)
	Teacher 9	5	5	0 (0%)
1ST SEM SY 2017-2018	Teacher 1	15	15	0 (0%)
	Teacher 2	12	14	-2 (14.29% ↓)
	Teacher 3	5	5	0 (0%)
	Teacher 4	12	15	-3 (20% ↓)
	Teacher 5	4	4	0 (0%)
	Teacher 6	20	20	0 (0%)
	Teacher 7	24	21	3 (14.29% ↑)
	Teacher 8	29	27	2 (7.41% ↑)
	Teacher 9	10	10	0 (0%)
2ND SEM SY 2016-2017	Teacher 1	19	22	-3 (13.64% ↓)
	Teacher 2	18	18	0 (0%)
	Teacher 3	7	7	0 (0%)
	Teacher 4	6	7	-1 (14.29% ↓)
	Teacher 5	4	4	0 (0%)
	Teacher 6	22	22	0 (0%)
	Teacher 7	24	21	3 (14.29% ↑)
	Teacher 8	26	25	1 (4% ↑)
	Teacher 9	10	10	0 (0%)

Table 2. Result of the comparison between the manual and automated distribution of unit per teacher

Scheduling	N	MEAN	ST. D.	Sig.	Remarks	Interpretation
Manual	9	15.11	8.038	0.865	Accept Ho	No Significant Difference
Automated		15.11	8.418			

4. CONCLUSION

The system comprising of five agents solves the problem of minimizing if not eliminating the hardware constraints. It also satisfies the soft constraint in handling class timetabling. The important feature of the system is the mapping of the agent autonomy into five agents to communicate effectively with each other in order to produce an efficient timetable for the university. The use of Prometheus platform helps in coming up with a detailed design of the system. It provides a detailed guidance in process and notations. It is not intended to be prescriptive, but it is developed in activities focusing on the use of multi-agent system.

Class scheduling is a coordination and combinatorial problem of finding an appropriate timetable for each course to be scheduled while avoiding conflicts simultaneously. The mathematical model used helps in the optimization process, specifically the rooms, faculty schedule, and student schedule. The course scheduling problem (CSP) solution model provides modularity to employ different algorithms in the development of the automated class timetable. In the evaluation of the effectiveness of the system, the comparison between the manually created class schedule and the generated automated schedule resulted in changes in the schedule and faculty loads due to the different conditions like faculty specialization and faculty time preference introduced in the system.

The study, therefore, concludes that the scheduling of classes in higher educational institutions can be automated by developing a multiple agent class timetabling system using Prometheus platform. The development of a multiple agent class timetabling system that automates the process of class scheduling of higher educational institutions (HEIs) using Prometheus Platform is open for further improvement. Based on the findings and conclusion of this study, it is recommended that: i) A more intensive analysis of performance differences across the set of instances should be carried out in order to find out which variables or features of the instances cause these differences. This should help to bring the principles into line with the realistic features of the timetable; ii) More real-world constraints may be added to the model. One idea is to coordinate challenging constraints by applying an ordinal scale (must, perhaps, want to increase system flexibility and prevent dispute over shared courses; students are able to take more classes. The chosen date-timeslots based on the needs of teachers and the shortest paths (distance between any two consecutive classes) within the university can also be generalized and integrated. The more constraints the model has the better it can be adapted to the needs of the real world; and iii) A student agent will be added to the conceptual framework in order to make the process more effective. The student agent will complete the class schedule by listing the students who will enrol for each subject.

ACKNOWLEDGEMENTS

The researchers would like to thank Dr. Dennis Gonzales who provided his insight and expertise to the success of this research paper. Also, the researchers would like to thank the panelists for sharing their pearls of wisdom and who are always willing to help in the improvement of this paper.

REFERENCES

[1] D. Magni, B. Orlando, and M. Del Giudice, "Exploiting Digital Skills in Higher Education: A Case Study Analysis," *IT and the Development of Digital Skills and Competencies in Education,* pp.1-20. 2021, doi: 10.4018/978-1-7998-4972-8.ch001.
[2] D. Magni and A. Sestino, "Students' learning outcomes and satisfaction. An investigation of knowledge transfer during social distancing policies," *International Journal of Learning and Intellectual Capital,* vol. 1, no. 1, p: 1. 2021, doi: 10.1504/IJLIC.2021.10035251.
[3] S. A. D. Aly and H. M. S. Badoor, "Performance evaluation of a multi-agent system using fuzzy model," *2018 First International Workshop on Deep and Representation Learning (IWDRL)*, 2018, pp. 7-12, doi: 10.1109/IWDRL.2018.8358208.
[4] H. Babei, J. Karimpour and A. Hadidi. "A Survey of Approaches For University Course Timetabling Problem," *Computes & Industrial Engineering*, vol. 86, pp. 43-59, 2015, doi: https://doi.org/10.1016/j.cie.2014.11.010.
[5] K.Y. Junn, J. H. Obit, R. Alfred, and J. Bolongkikit, "A Formal Model of Multi-agent System for University Course Timetabling Problems," *In book: Computational Science and Technology*, pp. 215-225, 2019, doi: https://doi.org/10.1007/978-981-13-2622-6_22.
[6] S. Srinivasan, J. Singh, and V. Kumar, "Multi-Agent Based Decision Support System Using Data Mining and Case Based Reasoning," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 4, p: 340, 2011.
[7] Y. Yang, R. Paranjape, L. Benedicenti, and N. Reed, "A System Model for University Course Timetabling using Mobile Agents," *Multi-gent and Grid Systems*, vol. 2, no. 3, pp. 267–275, 2006, doi: 10.3233/MGS-2006-2305.
[8] M. H. Abdalla, J. H. Obit, R. Alfred, and J. Bolongkikit, "Agent based integer programming framework for solving real-life curriculum-based university course timetabling," *Computational Science and Technology,* pp: 67-76, 2019, doi*:* 10.1007/978-981-13-2622-6_7.
[9] M. El-Helly, Y. Abdelhamid, M. Al-Wakeel, "A Mult-Agent Pattern Based Timetabling System," *Egyptian ComputerScience Journal*, vol. 35, no. 3, 2011.
[10] N. Oluwole, N. Oghenefego, and O. Obi, "Intelligent based Multi-Agent Approach for University Timetable Scheduling System," *International Journal of Computer Applications*, vol. 182, no. 1, pp. 10-21, July 2018.
[11] E. Burke, K. Jackson, J. H. Kingston and R. Weare, "Automated University Timetabling: The State of the Art," in *The Computer Journal*, vol. 40, no. 9, pp. 565-571, Jan. 1997, doi: 10.1093/comjnl/40.9.565.
[12] Z. Houhamdi, B. Athamena, R. Abuzaineddin, and M. Muhairat, "A Multi-Agent System for Course Timetable Generation," *TEM Journal*, vol. 8, no. 1, pp. 211-221, 2019, doi: 10.18421/TEM81-30.
[13] K. O. Babaagba, and S. A. Arekete, "A Design of an Agent Based System for Timetabling," *International Journal of Engineering Development Research*, vol. 5, no. 2, pp. 1168-1175, 2017.
[14] R. C. Richey, J. D. Klein, and W. A. Nelson, "Developmental Research: Studies of Instructional Design and Development," in *D. H. Jonassen (Ed.), Handbook of research on educational communications and technology*, pp. 1099-1130, 2003, doi: 10.4324/9781410609519-54.
[15] M. Wooldridge, *An Introduction to MultiAgent Systems*, New York, USA: John Wiley & Sons, 2009.
[16] D. R. Dos Santos, M. B. Ribeiro, and R. M. Bastos, "A Comparative Study of MultiAgent Systems Development Methodologies," *Second Workshop on Software Engineering for Agent-Oriented Systems*, pp: 37-48. 2012.
[17] L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems: A Practical Guide*, New York, USA: John Wiley & Sons, Jun. 2004, doi: 10.1002/0470861223
[18] J. H. Obit, "Developing Novel Meta-Heuristic, Hyper-Heuristic andCooperative Search for Course Timetabling Problems," Ph.D. Dissertation, School of Computer Science University of Nottingham, Nottingham, UK, 2013.
[19] R. Lewis, B. Paechter, and O. Rossi-Doria, "Metaheuristics For University Course Timetabling," in *book: Evolutionary Scheduling,* pp. 237-272, 1970, doi: 10.1007/978-3-540-48584-1_9

[20] S. Petrovic, Sanja, and E. Burke, “University timetabling,” *Conference Handbook of Scheduling: Algorithms Models, and Performance Analysis, chapter 45*, Jan. 2004.
[21] P. De Causmaecker, P. Demeester, and G. Vanden Berghe, “Evaluation of the University Course Timetabling Problem with the Linear Numberings Method,” *Proceedings of the 25th workshop of the UK planning and scheduling special interest group*, 2006, pp. 154-155.
[22] A. Muklason, R. G. Irianti, and A. Marom, “Automated Course Timetabling Optimization Using Tabu-Variable Neighborhood Search Based Hyper-Heuristic Algorithm,” *Procedia Computer Science,* vol. 161 pp 656–664, 2019, doi: https://doi.org/10.1016/j.procs.2019.11.169.
[23] Y. Yang, and R. Paranjape, “A Multi-agent System for Course Timetabling,” *Intelligent Decision Technologies Computer Science and Artificial Intelligence*, vol. 5, no. 2, pp.113–131, 2011, doi: 10.3233/IDT-2011-0101.
[24] A. M. Hambali, Y. A. Olasupo, and M. Dalhatu, “Automated University Lecture Timetable using Heuristic Approach,” *Nigerian Journal of Technology (NIJOTECH),* vol 39 pp 1-14. 2020, doi: 10.4314/njt.v39i1.1.
[25] S. Sen, “Solving the Course Scheduling Problem,” Master’s thesis, San Diego State University, 2012

BIOGRAPHIES OF AUTHORS

Dr. Angelita D. Guia is the current Associate Dean for Engineering Programs of the College of Engineering University of the East-Caloocan Campus. She is also a faculty member under the Computer Engineering Department. She is a graduate of Doctor of Information Technology and Master of Information Management at University of the East Manila. She is a Registered Electronics Engineer and a member of Institute of Electronics Engineers of the Philippines, Inc. His publication and research areas include environmental research, waste management, artificial intelligence, data analytics and deep learning.

Dr. Melvin A. Ballera is a graduate and one of the pioneers of Doctor of Philosophy in Computer Science. He is one of the Director of Research and Development at the Technological Institute of the Philippines. His publication and research areas include Natural Language Processing, Deep Learning, and Data Analytics.