THINK INDIA JOURNAL

ISSN:0971-1260
Vol-22- Issue-14-December-2019

An Architecture for Mobile Agent for the Polytechnic Course Timetabling System

Amosa Babalola, Ekuewa Jacob, Onyeka Ndidi, Olatunji Funso and Adeyanju Tosin.
Department of Computer Science,Federal Polytechnic, Ede,
Osun state, Nigeria

Abstract

Mobile agent technology represents a distributed computing paradigm that has been proposed as a means for increasing the efficiency of data aggregation tasks and addressing the scalability problem of centralized methods. A Course timetabling problem consists of scheduling a sequence of lectures between teachers and students in a prefixed period-of-time, satisfying a set of constraints of various types. Polytechnic Course Timetabling Problem is a hybrid optimization problem in the class of Non-Probabilistic hard problems. The aim of this research is *to design a mobile agent system that generates a Course timetable for the Polytechnic system. Java programming language in conjunction with the MySQL database was used to develop the resulting system. The resulting system of the design is a Polytechnic course timetable system that abides by all the constraints of a standard Course Timetable.*

Keywords: *Mobile agent, Course timetabling problem, Polytechnic system, Course timetable.*

1 Introduction.

Mobile agent technology represents a distributed computing paradigm that has been proposed as a means for increasing the efficiency of data aggregation tasks and addressing the scalability problem of centralized methods. Nevertheless, the itineraries followed by traveling mobile agents largely determine the overall performance of the data aggregation applications. Mobile agents have been proposed as a middleware solution for implementing efficient data aggregation schemes [1]. A mobile agent is an autonomous program moving from node to node and acting on behalf of the users toward the completion of an assigned task [2]. The mobile agent paradigm is an emerging technology that promises many benefits in network computing. A mobile agent is a program that can migrate from one computer to another for remote execution. A mobile agent has the following properties: state, implementation, interactions and a unique identifier [3]. The state abstracts the agent functionality and represents the agent in the system. Agent implementation denotes its behavior and interactions enable the agent to reason about its environment and to co-operate. A unique identifier is needed to address the agent globally. The agent can also have metadata related to its operation and interactions, such as security and privacy attributes. Mobile agents are expected to be able to roam over heterogeneous networks such as the Internet.

Course timetabling is a difficult optimization problem which involves the process of assigning times and rooms to classes, creating a course timetable for students while respecting various restrictions and preferences including Courses: size, room equipment, structure; Instructors: availability, preferred times; Students: curricula, pre-registrations and Others: number of rooms available and their sizes[4] . The definition timetabling is often used when solving problems with some form of personnel allocation, such as Polytechnic Course Timetabling. Due to the personnel allocations, many people are affected by the timetables created. For the combined planning problems in the educational system, four main stakeholders are identified.

a. The administration: solves the planning problems and sets the minimum standards

Copyright © 2019Authors

ISSN:0971-1260
Vol-22- Issue-14-December-2019

that the timetables must conform to, the capacity of the room, consecutive exams, etc.

b. The students: This is by far the largest in the volume of the four stakeholders. The students seldom have much influence on the outcome of the different planning problems. And due to the number of students involved it is difficult to declare what is the best timetable for the students as they all might have different wants and needs.

c. The teachers: The teachers have a little more influence on the outcome than the students and they do also request a compressed timetable. Furthermore, they might have some restrictions on which day and time they want to give lessons or exams.

d. The departments/sections: Often only relevant for polytechnics as the high schools rarely are organized with departments or sections. The department can have some restrictions or requests for the exam and course timetabling.

For all the various planning problems some restrictions always apply, these are the so-called "First-order conflicts". For example, no person can be at more than one place at any time. These first-order constraints are always present in some form. Besides these constraints, there is a great variation between the problems, but also within a given problem there is great variation due to different polytechnics, high schools and/or the Ministries of Education of the respective countries.

Polytechnic Course Timetabling Problem is a hybrid optimization problem in the class of Non-Probabilistic hard problems [5]. A Course timetabling problem consists of scheduling a sequence of lectures between teachers and students in a prefixed period-of-time, satisfying a set of constraints of various types. It is the weekly scheduling for all the lectures of a set of Polytechnic courses, removing the overlaps of lectures of courses having common students. The primary objective in preparing timetable is obtaining a conflict-free schedule for each activity sharing the same resources.

With this current problem there a need to develop an agent based system for creation of timetables to ease the problems associated with such. The mobile agent will be able to search through a database and gather the necessary information needed for the creation of the timetable. The mobile agent system will then, based on the information gathered, generate a timetable for the Polytechnic. The agent will be able to jump from one database to another if the information gathered for the creation of the timetable is incomplete. The aim of the research is to design a mobile agent system that generates a Course timetable for the Polytechnic system.

2.0 Review of Related Works.

An Agent is a software element that is autonomous with intelligence, mobility while having the ability to collaborate well by other users, Intelligent Agents, and Systems. They hold responsibilities without constantly consulting the user with a dedication, on their behalf. They can effectively co-ordinate among agents, users and software hosts. A searching agent that is competent to retrieve preferred information on behalf of the users. The agent is sent away to carry out the look for off-line as well, after being told what to search for. Eventually, the agent can move back with needed output [6].

MA paradigm is an extension of client/server and remote evaluation paradigms that provides a new development way for distributed systems. Mobility is the novel concept that characterizes the paradigm. It is the migration capability of an agent from a host to another one where it resumes its execution to achieve a task. They have a heterogeneous nature, operate asynchronously, adapt dynamically and react autonomously to changes Furthermore, MAs reduce the network load and overcome its latency by acting locally. They are also robust, fault-tolerant and they encapsulate communication protocols that facilitate their upgrading. Despite these notable advantages, the paradigm brings significant problems especially security threats which have arisen because a mobile code generated by one party will move to and execute in an environment controlled by another party. This is posing multiple issues that must be taken into account, including authentication, authorization (or access control), intrusion detection, etc [7].

A mobile agent represents a program capable of migrating from one node to another in a network to perform certain designated tasks. The ability to

Copyright © 2019Authors

ISSN:0971-1260
Vol-22- Issue-14-December-2019

migrate code and processing functions to a remote node offers the potential benefits of reduced network traffic and bandwidth requirements. The use of mobile agents for network application management has been proposed and investigated by several researchers in recent years with the primary goal of reducing network traffic and building scalable systems. A Mobile Agent (MA) is a composition of computer software and data which can migrate (move) from one computer to another autonomously and continue its execution on the destination computer [8].

Mobile Agents are the programs that move between computers or nodes of a network, autonomously trying to fulfill some specific goals given by users. Agents are different from other applications in that they are goal-oriented: they represent users and act on their behalf to achieve some set goals autonomously – i.e. they control themselves, as in the decision where and when they will move to the next computer or node. Mobile Agents do provide a viable means of performing a network security assessment and analysis efficiently and effectively. Also, the mobile-agent is an execution unit that can migrate autonomously to another host and resume execution there, continuing from where it left off [9].

Mobile agents are software abstractions that can migrate across the network, suspend action there and migrate from there to another node, and resume operations from where they left off on the previous node. It has the propensity of communicating in agent language. In concept, a mobile agent can move its complete virtual machine from one host to another. While, on a particular host, the mobile agent can carry out some tasks, and afterward, it can migrate to another host to continue operation from where it stopped action on the previous host. As an executing program, a mobile agent is made up of code, data and execution state and is embedded with some intelligence and the ability to autonomously migrate across the network. As it moves, the mobile agent can represent its owner in various tasks and satisfy its requests. It can communicate with the host as well as other agents on the host; it can sense its environment and carry out a set of activities to attain its mission and achieve its goals [10].

[11] proposed Agent UML (AUML) which can be considered as the best-known UML-based extension for agents. AUML offers the possibility of describing the internal behavior of an agent as well as its external behavior. The core parts of AUML are agent class diagrams and interaction protocol diagrams which extend respectively UML class diagrams and sequence diagrams. It was initially developed to deal only with multi-agent systems, but its deployment and activity diagrams were extended thereafter [12] and [13], to support modeling of the agent mobility. The novel notations introduced by the AUML deployment diagram extension facilitate the modeling of an agent migration to different nodes and the locations concerned by the move. Furthermore, the issue of time can be expressed by AUML activity diagrams, i.e. when an MA has to migrate. Multiple concepts and notations have been developed in this language [14], but the essential features of MA-based applications are not taken into account. The mobility, for example, is statically described and the security is entirely omitted.

[15] and [16] introduce Agent Modeling Language (AML); an extension of UML that incorporates different concepts and features of multi-agent systems. AML is currently considered as the most complete published agent-oriented modeling language. It covers the modeling of the static structure, dynamics, and behavior of entities in a multi-agent system. AML attempts also to provide modeling support for mobility by recognizing and identifying the properties of the agent environment, host, and comportment (move and clone). The authors consider mobility only at the implementation level when the multi-agent system is deployed. This makes it unsuitable for modeling real MA-based applications.

Polytechnic Course Timetabling problem is the task of assigning lectures of courses to time slots, rooms and other resources, subject to constraints applied for the single student. The large size and structure of polytechnics often make the problem of solving the course timetabling so complicated that it is solved by several experts scattered across different faculties and departments. Due to the complexity of the problem, it is quite common that the polytechnics re-use the solution from previous years when solving the problem.

The Polytechnic Course Timetabling is often divided into two general approaches; the Curriculum-based Polytechnic Course Timetabling and the Enrollment-based Polytechnic Course Timetabling. Courses that can be taken in combination, because they are needed to satisfy the degree rules of the given study, are called curricula. And assigning

Copyright © 2019Authors

ISSN:0971-1260
Vol-22- Issue-14-December-2019

courses to time slots based on this is called Curriculum-based Course Timetabling.

The problem consists of creating the weekly schedules of lectures where conflicts between the courses are set accordingly to the curriculum specified by the Polytechnic. The second approach is the Post Enrollment-based Course Timetabling. Based on the enrollment data of each student, it is determined where courses are placed in the timetable such that all students can attend the events on which they are enrolled. This approach has been given various names in the literature including the Class Timetabling Problem and Event Timetabling. The papers that worked on the Post Enrollment-based Course Timetabling include [17] and [18]. The two approached are not necessarily mutually exclusive. The polytechnics can use the curriculum-based approach in an early stage to get the basic structure of the timetable, and then, later on, use the enrollment-based approach to improve the timetable.

3 The Architecture of Mobile Agent for Timetabling for the Polytechnic System

The mobile agent environment comprises a central computer and some databases. The databases reside on a typical computer which is composed of some hardware devices such as CPU, memory and so on, in which an operating system is running. Other categories of software can be running on the computer. This includes front-end software, back-end software, and utility software. The architecture of the proposed mobile agent indicates that when the user searches for information from Database 1, it stores the information of available lecturers and associated courses if otherwise, it will automatically move to Database 2 and repeat the same process until it reaches the last Database n. When the data have been found, it will now report back to the central computer showing the information about the lecturers and courses found in the right category. The way the system operates is highlighted here with the migration agent (information retrieval agent) at work. The mobile agent environment is presented in Figure 1

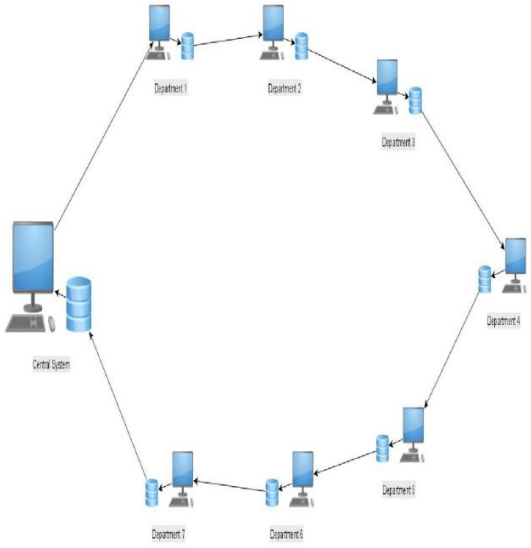

Figure 1: System Architecture

The architecture of the proposed mobile agent is conceptualized in Figure 1. The platform is for the take-off of the Agent Monitor at the central computer and the platform for its landing at the target databases. At the central computer, the Agent Monitor is created and equipped with the code, data and other necessary parameters and dispatched to the target databases. The Agent Monitor then navigate autonomously through the databases from the central computer and interact with the host operating system of the target databases and its utility programs as it retrieved the desired information. The Agent Monitor moves from one database to another while carrying along with intermediate results.

Copyright © 2019 Authors

ISSN:0971-1260
Vol-22- Issue-14-December-2019

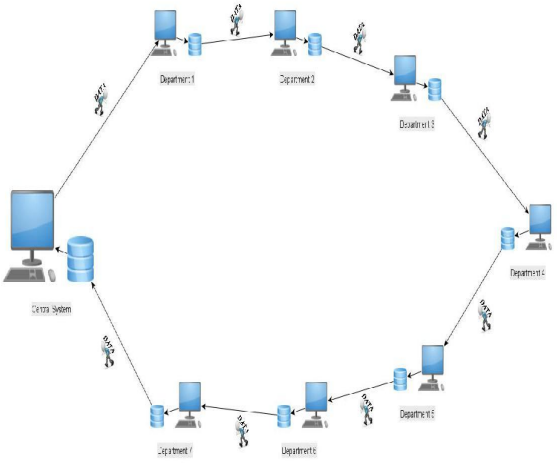

Figure 2 Agent Mobility of the System

The results obtained by the mobile agent after successful visits to a set of target databases are transferred to another database, which is displayed on its screen or printed out for external analysis, interpretation, policy formulation and decision making by the User. The design of the Agent Monitor has three major components namely:

a. Inputs: the input to the Agent monitor is the target machine identity,
b. Agent Monitor, and
c. Output report.

4. Development of Mobile Agent for Polytechnic Course timetabling

In the retrieval of information from a database, there is the main issue to resolve. The system has to retrieve the information from the specified databases and store them in another database for processing on the central computer. Hence, we have a mobile agent (agent monitor). The agent is initiated to make the proposed system as presented in Figure 2. The Agent Monitor is a mobile agent dispatched from the central computer with its code, data and other parameters to other databases in the system. The major function of the Agent Monitor is to go into the database and retrieve the related information in the databases whose identities are known. At each database, the Agent Monitor stores the information gotten at each database into another database which it carries along with it for the collection of information about the lecturer and associated courses. The information collected at each database is placed in the database container of the Agent Monitor for onward movement to the next database. Program flowcharts showing the flow of the agent in the system is in. Figure 3.

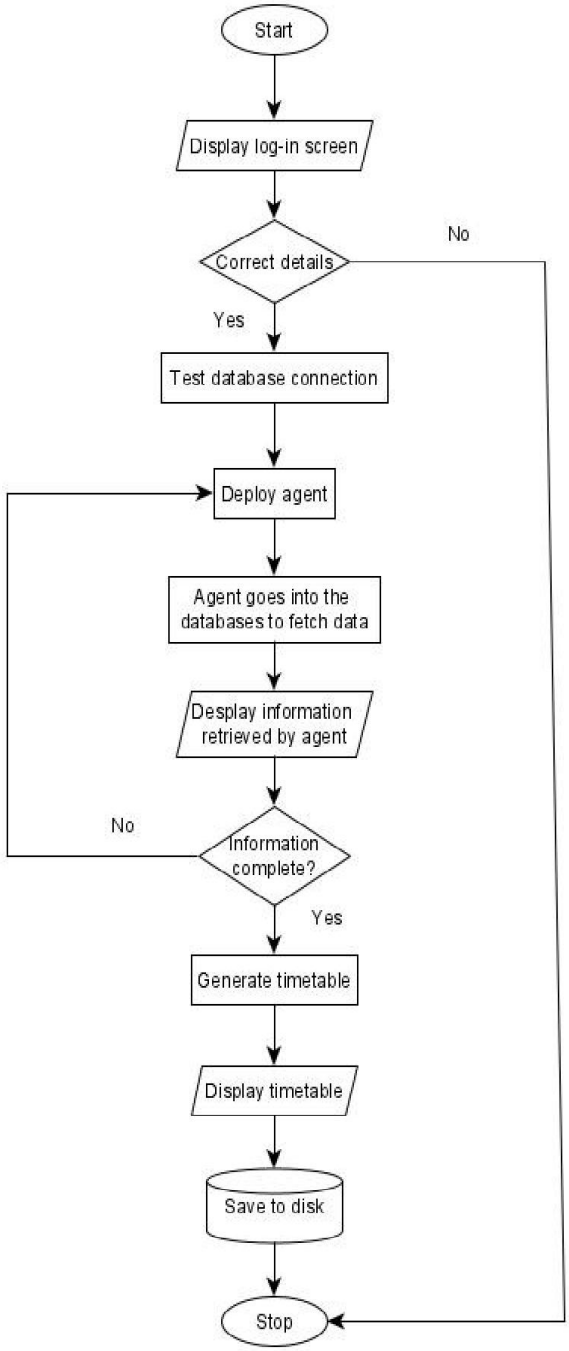

Figure 3: Agent flow in the System

Copyright © 2019 Authors

ISSN:0971-1260
Vol-22- Issue-14-December-2019

The information retrieved from the database is verified to certify that the retrieval agent can collect all the necessary information. The basic information student has to input and submit to the system which enables the system to register the student for the timetable creation process are: Course code, Course-name, Lecturer identification number, Lecturer name

The program was developed using the Java programming language as already mentioned. The Department of Computer Science, Federal Polytechnic Ede was used this research. For any desktop computer to run a Java application program the most basic requirement will be to install a version Java Development Kit (JDK). For the purpose of this research we use JDK 8 for simplicity of purpose.

The resulting system test results states exclusively the output gotten from the implementation and testing of the Mobile agent system on a computer system using Eclipse IDE as an Integrated Development Environment. The system was tested with various input device, valid test data were used to test the workability, reliability, and efficiency of the system. Some of the interfaces obtained from the system during testing are as shown in Figures 4 and 5 respectively.

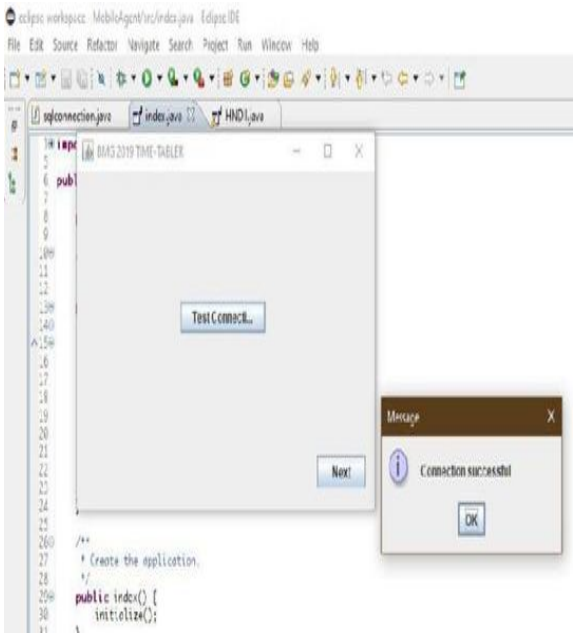

Fig. 4: Connection test page

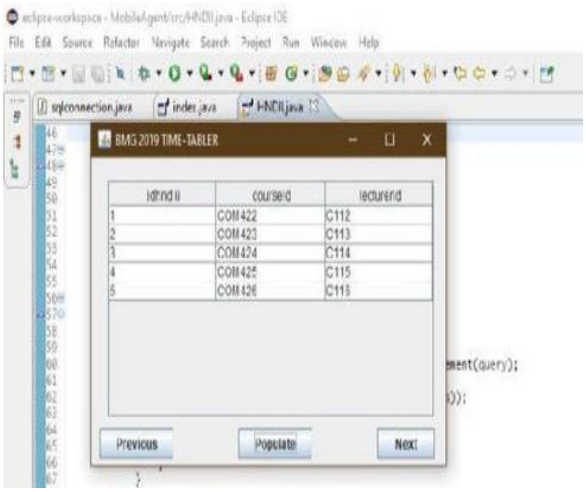

Fig. 5: Information Retrieval page

5. Conclusion

The role of ICT in production can be seen in three ways: as a tool, as an alternative, or as an agent of change. Policy-makers have often tended to view it in the first two guises: either as a supplement to existing services or a potential substitute for such services. In this study, an Agent-based system for creating Course Timetables in Polytechnics has been presented. The records are protected and stored safely through the use of a password preventing an unauthorized person from intrusion. The timetables can also be easily accessed, and also printed if need be for the activities of the decision-makers. With the advent of IT and an increasingly diverse, new mobile technologies and expanding opportunities it will be regarded as retardation to keep to the manual ways of doing things. The provision of ICT facilities in the laboratories and libraries presupposes that lecturers and students alike can use them to improve awareness, adjustment, and learning. Investment in ICT facilities will help in counseling and other support services necessary for the effective delivery of an ICT-based curriculum and this should be utmost in government priorities.

The importance of this system cannot be overemphasized and it eliminates redundancy of data due to RDBMS installation and also reduces time and effort wasted in preparing course timetables. The manpower and energy needed in the manual system can be directed to another facet of importance within the Education system. It is hereby recommended that a more detailed study should still be made in case of

Copyright © 2019 Authors

ISSN:0971-1260
Vol-22- Issue-14-December-2019

a change in the system. Further works on Mobile Agent for Polytechnic Course Timetabling is planned to be in the three agent phases i.e. Retrieval agent, Database agent, and Timetabling agent. The advent of API and Graph QL is also advancement in distributed computing technology and should be considered in the development of distributed systems.

References

[i] Dong, M., Ota, K., Yang, L.T., Chang, S., Zhu, H. & Zhou, Z. (2014). Mobile agent-based energy-aware and centric data collection in wireless sensor networks. Computer networks, 74: pp 58-70.

[ii] Amosa, B. M. G., Ekuewa, J.B., Oyetunji, O.O., Nwaekpe, C. & Ogunleye, T. (2017). Operability of Mobile Agent Applications in a Protected Environment. IJFRCSCE ISSN: 2454-4248. Vol. 3 Issue: 12. pp 190.

[iii] Teemu, L., Jose, A. L., Yoshito, T., Kaoru, S. & Jukka, R. (2015). Mobile crowdsensing with mobile agents. DOI 10.1007/s10458-015-9311-7. pp 12.

[iv] Zuzana, M. & Tomas, M. (2015). Course Timetabling With UniTime. Apereo Webinar Series September - December 2015. pp 4.

[v] Babaei, H. (2015). A Review of Distributed Multi-Agent Systems Approach to Solve University Course Timetabling Problem. Advances in Computer Science: an International Journal, 3(5). pp 19-28.

[vi] Belghiat A. (2015). Bridging the Gap between Modeling of Mobile Agent-based Systems and Semantic Web using Meta-Modeling and Graph Grammars. In The 7th International Conference on Information Technology (ICIT), Jordan. Pp 195-197.

[vii] Birukou, A., Blanzieri, E. & Giorgini, P. (2012). "Implicit: a multi-agent recommendation system for web search," Autonomous Agents and Multi-Agent Systems, vol.24, no.1, pp.141–174.

[viii] Amosa, B. M. G., Adejuwon, A. O., Longe, O. L., Onyeka, N.C. & Abasiene, J.U. (2018). "Mobile Agent System for Handling Criminal Cases " International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 3, Issue 3, ISSN : 2456-3307. pp 170-171.

[ix] Amosa B.; Onyeka N.; Olaniyi B. Babafemi O. (2017). "Mobile Agent for Monitoring and Evaluation of Security Applications in a Network Environment". International Journal of Computer Science and Network, Volume 6, Issue 6, pp 826.

[x] Arekete, S. A. (2013). "Development of Mobile Agent for Monitoring and Evaluation of User Activities in a Network Environment", Ph.D. thesis, Department of Computer Science, Federal University of Technology, Akure, Nigeria, 2013. Pp 46-49.

[xi] Bauer B. (2001). Agent UML: a formalism for specifying multi-agent interaction. In 22nd International Conference on Software Engineering (ICSE), Agent-Oriented Software Engineering, Springer, Berlin, pp. 91–103.

[xii] Mouratidis, H. (2002). Extending the Unified Modeling Language to Model Mobile Agents. In Proceedings Agent Oriented Methodologies Workshop, Annual ACM Conference on Object Oriented Programming, Systems, Languages (OOPSLA), Seattle, USA. Pp 78-95.

[xiii] Poggi, A. (2004). Modeling Deployment and Mobility Issues in Multiagent Systems using AUML. In Agent Oriented Software Engineering IV, P. Giorgini, J. P. Muller, J. Odell Eds. LNCS 2935, Springer-Verlag. Pp 57.

[xiv] Bauer B. & Odell, J. (2005). UML 2.0 and agents: how to build agent-based systems with the new UML standard. Engineering Applications of Artificial Intelligence, vol. 18. Pp 141–157.

[xv] Cervenka, R. & Trencansky, I. (2007). The Agent Modeling Language – AML: A Comprehensive Approach to Modeling Multi-Agent Systems. Springer Science & Business Media. Pp 102-126.

[xvi] Cervenka R. (2005). AML: Agent Modeling Language Toward Industry-Grade Agent-Based Modeling. In Agent-Oriented Software Engineering V: 5th International Workshop, J. Odell, P. Gioginin, and J. P. Muller, Eds. Springer-Verlag, Berlin. Pp 31–46.

[xvii] Ceschia, S., Di Gaspero, L. & Schaerf A. (2012). Design, engineering, and experimental analysis of a simulated annealing approach to the post enrolment course timetabling problem. Computers & Operations Research, 39(7):1615 1624. ISSN 0305-0548. Pp 567.

[xviii] Nothegger, C., Mayer, A., Chwatal, A. & Raidl, G. R. (2012). Solving the post enrolment course timetabling problem by ant colony optimization. Annals of Operations Research, 194(1): ISSN 0254-5330. Pp 325-339

 Copyright © 2019Authors