Multi-agent Based University Time Table Scheduling System (MUTSS)

[1]M. P. Tariq, [2]M. Waqar Mirza and [3]R. Akbar
[1,2]Department of Computer Science, Virtual University, Lahore-Pakistan
[3]Department of Computer and Information Sciences, UTP - Malaysia

Abstract - **This paper presents a multi-agent based approach for the development of time table scheduling system for educational institutes. The information capturing and saving agent captures data from user and saves it into the database. The information publishing agent publishes information on the web site or sends to the users' inboxes. The environment monitoring and processing agent monitors the environment. An algorithm for the development of time table is presented which shows the collaboration and cooperation of the three agents.**

Keywords - Multi-agent, Intelligent agents, Time table,
Course scheduling

I. INTRODUCTION

Currently, in almost all educational institutes, there is no reliable or an efficient intelligent time table system [1]. It is a reality that all educational institutes ranging from small to large ones are following the manual system which is a very difficult and time consuming task. This manual system becomes a hot issue at the start of each new semester for the head of department (HoD) [2]. We can not get historical data to make the job easy and comfortable. Among other traditional problems of a manual system like lack of accuracy, slow speed of the system and problems in sharing of information etc. the key problems are to: 1) Keep and manage record of the previous data. 2) Meet in person. 3) Manage multiple queries for the same subject. 4) Make availability of the interested subjects for a faculty member. 5) Mange rooms for delivering lectures. 6) Have insufficient information by HoD during the development of time table. Therefore, to design and develop a time table for educational institutes is not an easy and comfortable job.

In this paper, authors propose a Multi-agent based University Time Table Scheduling System (MUTSS) as a solution to this problem. MUTSS, which is a web based system, assists administrative persons like chairman, head of department or director of an educational institute in developing a very accurate time table without face to face interaction with other concerning persons. Working of the proposed system depends upon three agents. The information capturing and saving agent (ICSA) is responsible to capture data from a user interface/web page and saves it into the database. It changes the environment.

The environment monitoring and data processing agent (EMPA) monitors the environment, performs computation, and asks IPA to publish it on the web site or send it to the users. The information publishing agent (IPA) publishes data on the web site and/or sends it to the users as per instructions of EMPA. In addition to its core job (time table development), the MUTSS also provides a number of features like defining and publishing of alerts, notifications, and announcements, viewing and editing profiles, setting of course(s) etc.

In the past, several systems were proposed for the development of time table. Some of them have the lack of intelligence approach and some are desktop systems while some are not implemented. Oprea [1] proposes a multi-agent based system to schedule university courses. He presents architecture of the system and shows communication of the agents via various communication diagrams. However, validation of the proposed system is not a presented. Fang2 presents a UML based course scheduling system. Its limitations are that it is neither an intelligent system nor it can be accessed from around the world. System proposed by Takegami and Takegami [8] is for commercial institutes and uses spread sheets for processing. It is also neither an intelligent system nor it is accessible from any where in the world. Ho et al., [9] present an algorithm to solve the problem of developing the time table but it also lacks intelligent based approach and accessibility from around the world. The proposed system covers all of the problems described above. The major features of our system are that firstly it is an intelligent system secondly it is a web based system and can be accessed from any part of the world.

The article is organized as follows: In section 2, we present materials and methods of the proposed system. In section 3, we provide results and discussions. In section 4, we give concluding remarks and future directions.

II. MATERIALS AND METHODS

In this section, first we provide architecture of the proposed system. Then a mechanism to conduct communication among the agents is described and at the end of this section, we describe whole process to generate time table by collaboration and cooperation of the three agents.

A. System Architecture

Software architecture reveals the relationship among major components of the system. One of its major objectives is to reduce the development cost and establish the consensus among the developers and clients of the system [11][12]. In this section, we present two types of architecture of MUTSS: 1) General architecture (figure 1). 2) Three tier architecture (figure 2).

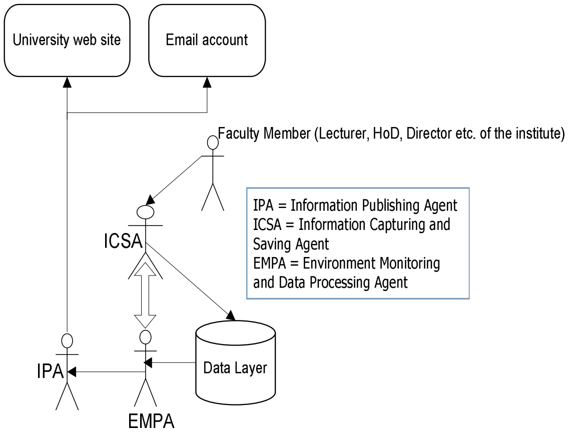

University web site
Email account
Faculty Member (Lecturer, HoD, Director etc. of the institute)
IPA = Information Publishing Agent
ICSA = Information Capturing and Saving Agent
EMPA = Environment Monitoring and Data Processing Agent
ICSA
IPA
EMPA
Data Layer

Fig. 1: General Architecture of MUTMS

1) General Architecture

Detail of each of the components of the general architecture is described as follows.

a) Information Capturing and Saving Agent (ICSA)

This agent captures data provided by a faculty member and/or head of the department and saves it into the database. Actually it changes the environment when it saves data in database on behalf of a user. As an example, environment changes when ICSA saves data on behalf of a faculty member relevant to course offering as shown in figure 5. Similarly ICSA also changes environment by saving data on behalf of HoD regarding defining alerts. It also interacts and cooperates with EMPA and sometimes saves data in database on the request of EMPA. Sometimes it may ask EMPA, on the behalf of a faculty member or head of department to perform some tasks like report generating and publishing other information. For example, HoD may ask ICSA to generate a report of instructors who teach the same subject to both BSCS and MSCS students. ICSA then forward this message to EMPA and EMPA collects desired data from database; gives it a shape of a report and finally it asks IPA to deliver it to the HoD.

b) Information publishing agent (IPA)

This agent publishes (on the web site) and sends notifications, announcements, alerts and other messages to the accounts or inboxes of both a faculty member and the head of department. It performs this job on the request of EMPA. For example, an alert "A meeting is going to be held on Oct. 22, 2009, for detail information check your inbox" is defined by HoD on Oct. 10, 2009 and EMPA is instructed to propagate this alert on Oct. 16, 2009. In this case ICSA saves this information in database. EMPA, as soon as, the said date i.e. Oct. 16, 2009 reaches, collaborate with and requests IPA to send this alert to all the specified faculty members.

c) Environment Monitoring and Data Processing Agent (EMPA)

This agent is the most important and core component of our system. It performs all business logic relevant tasks of our system. It monitors its environment around the clock and takes an immediate and appropriate action in case there is a change in the environment. Environment may be changed when data is saved in database by ICSA. ICSA may save data relevant to course(s) offering (figure 5) in database or when an alert is defined by HoD. This agent may take an action in two cases. Firstly, when it receives a request directly from HoD or a faculty member through ICSA to perform a job. Secondly, when it finds any change in the database i.e. its environment.

d) Data Layer

Fourth layer is the data layer which possesses data required by EMPA to perform various tasks. Actually this layer is an environment for the agents. When any type of change occurs in this layer, EMPA comes in action and performs an appropriate action. Change in this layer occurs when data is saved in this layer by ICSA. This layer possesses data organized into various tables with appropriate relationships. EMPA continuously observes this layer and acts if there is a change in this layer.

B. Three Tier Architecture

Figure 2 shows three tier architecture of the proposed system. Since, IPA is concerned with the publishing of data/information so it is placed in the presentation layer. Business logic layer of the MUTMS comprises of EMPA and ICSA. As shown by figure 2, ICSA gets data from presentation layer and saves it into the database. ICSA also collaborates with EMPA when HoD asks it to generate a report. EMPA collects data from data layer, convert it into the required format and asks IPA to publish it on the web site.

C. Communication among the Agents

Following is knowledge query manipulation language (KQML) [10] dialogue between ICSA and EMPA. Pictorial representation of this communication is shown by figure 3. This dialogue is initiated when HoD asks ICSA to do some task. For example, HoD queries ICSA to tell the instructor who teaches the subject of 'Operating Systems'. ICSA sends this query to EMPA using 'ask-one' KQML performative and EMPA replies back using 'tell' performative that Prof. Kinglay teaches the subject of

'Operating Systems'.

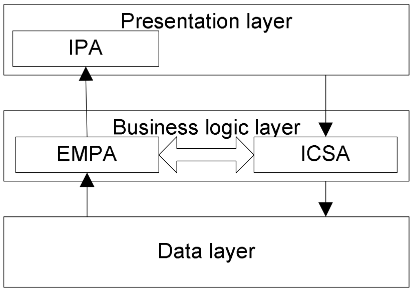


Fig. 2: Three Tier Architecture of MUTMS

1) KQML dialogue between ICSA and EMPA

(ask-one

:sender ICSA
:content "course-owner(operating systems, Course Owner)"
:receiver EMPA
:reply_with course-owner-query
:language Prolog
:ontology MA-TTS)

(tell

:sender EMPA
:content "course-owner (operating systems, Kinglay)"
:receiver ICSA
;in_reply_to course-owner-query
:language Prolog
:ontology MA-TTS)

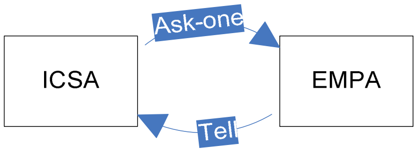


Fig. 3: Communication between ICSA and EMPA

D. Time table development Process

This process starts when HoD provides the following information to our system through ICSA and instructs to send an invitation to all concerned faculty members on a specific date and time. This invitation is a request to offer the courses along with day, time and available rooms in the current semester.

1. List of the courses to be offered in a semester.
2. Days on which the classes can be conducted. For example, the classes can be arranged from Monday to Friday. Saturday is only for other activities like seminar, faculty meeting etc.
3. Time duration during which classes can be scheduled. For example, classes can be conducted only from 8.00am to 3.00pm
4. Information about the rooms where classes can be conducted. For example, HoD specifies that rcs1, rcs2, and rcs3 are available. 'Rcs' stands for computer science room.

As EMPA continuously monitors the environment, therefore, when the specified date and time reaches, EMPA collects all information provided by HoD, gives it a proper format and requests IPA to send an email to all concerned faculty members requesting them to offer the course(s). The email provides a link to go to the page where the faculty members will select course(s), day time and room.

Faculty members go to the web page by clicking the link provided by IPA. The web page helps them to select the course(s) along with preferred day, time and venue. All information provided by a faculty member is saved in DB by ICSA. In case a clash occurs in course offering, day, time or room selection, EMPA takes a prompt action and informs the concerned faculty member by IPA about the clash and suggests him/her alternatives or refer them to HoD. EMPA follows the following rules during the provision of alternatives:

- HoD has the top level priority. So, if HoD selects a course, day, time or room which already has been selected by other faculty member, EMPA lets HoD select the course and informs the other faculty member about the situation and asks him/her to offer another course, day, time or room.
- Senior most faculty member has second level priority. This case is also handled like the case 1.
- In case of a subject clash, the subject will be assigned to the teacher who had taught the subject in any previous semester(s).
- In case of a room clash, the priority will be given to the teacher who had been taking classes in this room in the previous semester(s).
- In case of class timing clash, the priority will be given to the teacher who had been taking classes at the same time in the previous semester(s).

If a clash occurs among faculty members having the same seniority level, then EMPA asks IPA to send an email to all faculty members involved in creating the clash and requests them to have a meeting to resolve the issue. If clash still persists then EMPA informs HoD via IPA and asks to get resolved the clash. In case of any other clash which is not being resolved by the faculty members themselves, EMPA informs HoD via IPA and asks him to handle the matter and resolve it.

Similarly if a delay is being faced due to a faculty member in offering the course(s), EMPA sends him/her a reminder via IPA. If the faculty member does not offer the course even after receiving a reminder, EMPA prepares a message and asks IPA to send it to HoD. HoD then, may call him/her on his/her land line or cell number to ask him/her to offer the course(s).

When all faculty members have offered the courses, and there is no clash or delay, EMPA collects all this information gives it a shape of time table, forwards it to the

HoD for verification from him. If HoD does not verify time table and makes an objection, EMPA acts as per new instructions of HoD. For example, HoD may instruct EMPA to contact ‘facultymember1’ and ‘facultymember2’ and ask them to interchange their course(s). In this case, EMPA sends them an email and ask them to interchange course(s) as per instructions of HoD. If HoD verifies the time table, EMPA publishes it via IPA on the university web page and emails it to all faculty members for their information as well.

During the whole process of developing time table no person is involved. The whole process is automatic and intelligent enough to create a consistent and agreed time table. Figure 4 shows the complete process of creating the time table.

E. *Tools and Technologies*

The proposed system is developed using the following tools and technologies: PHP [3][4] is used as a computer language. MySql [5] is used as database. HTML [6] for presenting a cool and user friendly look to the user. JavaScript [8] is used for client side data validation and to support DHTML.

F. *Algorithm for development of time table*

The development of time table is shown in the following via ‘timetable’ algorithm. Our time table algorithm calls two algorithms to complete its execution. These algorithms are for ICSA and EMPA. As described earlier too, that the responsibility of ICSA is to get data/information from the user interface (a web page) and save it in database. Algorithm of ICSA shows the same activity. In first step it gets data provided by HoD regarding the course(s) which are to present to faculty members so that they can select their interested course(s). In second step, it saves data in database. In third step, it gets data about the course(s), rooms and time provided by a faculty member and then calls the step 2 to save it in the database. Algorithm of EMPA does the actual work. As soon as, data relevant to course(s) offering of a faculty member is saved in database by ICSA, EMPA comes in action. It checks for all types of clashes and delays. If a normal clash occurs, it prepares a message consisting of various alternatives and asks IPA to send the message to the concerned faculty member. For example, if a faculty member selects rcs1 (room) on Monday at 8.00am which has already been selected by a senior teacher, EMPA informs the faculty member that rcs1 is available on Monday only from 10.00am to 2.00pm so you should select this room on available day and time. If a clash occurs between faculty members having the same seniority level, then EMPA compiles a message instructing the relevant teachers to have a meeting and resolve the issue with the mutual consent. EMPA sends this to IPA to send to the concerned faculty members. I a sever clash occurs then EMPA prepares a message and request IPA to send it to HoD. It is a request to HoD to get involve and resolve the issue. Similarly, delay may also of two types. In case of a normal delay, EMPA sends a reminder to the concerned teacher via IPA. But in case of a severe delay, EMPA again refer the case to HoD. EMPA requests HoD to ask the concerned faculty member to come online and offer the course(s). At the end when all faculty members have entered data relevant to course(s) offering and there is no clash or delay, EMPA prepares time table and sends to the HoD for the purpose of verification. After verification from HoD, EMPA publishes the time table on web site and sends it to all concerned faculty members as well. This process has also been shown by figure 4.

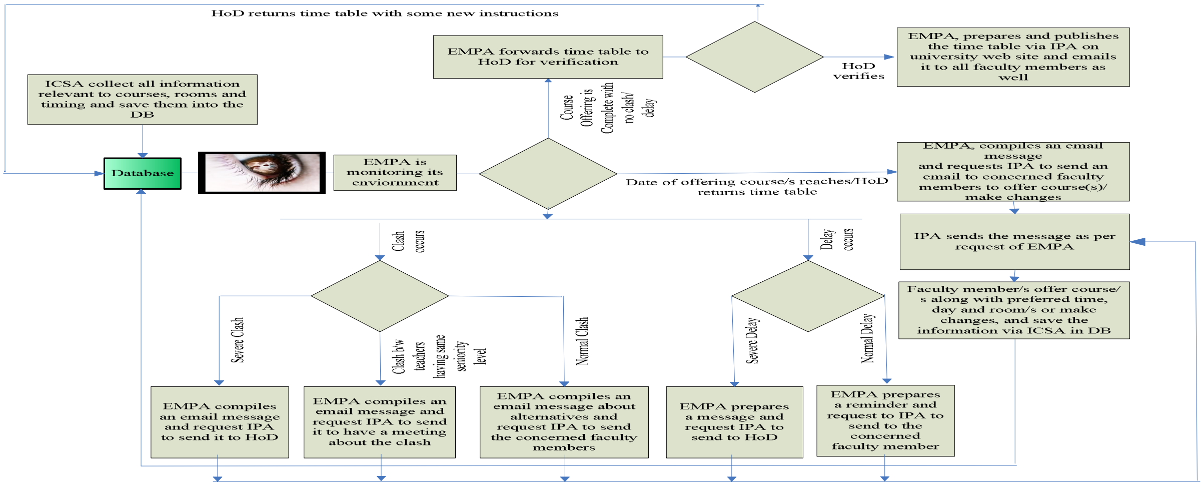

HoD returns time table with some new instructions
EMPA forwards time table to HoD for verification
HoD verifies
EMPA, prepares and publishes the time table via IPA on university web site and emails it to all faculty members as well
ICSA collect all information relevant to courses, rooms and timing and save them into the DB
Course Offering is Complete with no clash/ delay
Database
EMPA is monitoring its enviornment
Date of offering course/s reaches/HoD returns time table
EMPA, compiles an email message and requests IPA to send an email to concerned faculty members to offer course(s)/ make changes
IPA sends the message as per request of EMPA
Faculty member/s offer course/ s along with preferred time, day and room/s or make changes, and save the information via ICSA in DB
Clash occurs
Delay occurs
Severe Clash
Clash b/w teachers having same seniority level
Normal Clash
Severe Delay
Normal Delay
EMPA compiles an email message and request IPA to send it to HoD
EMPA compiles an email message and request IPA to send it to have a meeting about the clash
EMPA compiles an email message about alternatives and request IPA to send the concerned faculty members
EMPA prepares a message and request IPA to send to HoD
EMPA prepares a reminder and request to IPA to send to the concerned faculty member

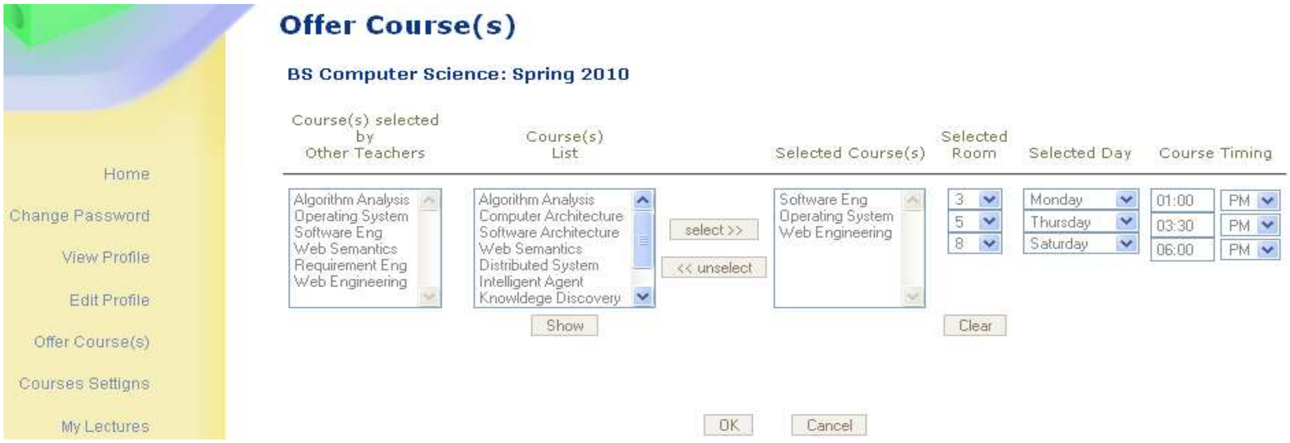
Fig. 4: Development of Time Table by Collaboration and Cooperation of the three Agents

```
Algorithm Timetable
        Call ICSA
        Call EMPA
End
```

```
Algorithm ICSA:
    1. Gets information relevant to course(s), rooms and timing from HoD
    2. Saves the information in database
    3. Gets data relevant to course(s) offering of a faculty member and GO TO 2.
End
Algorithm IPA:
    1. Get instructions from EMPA
    2. Publishes data on web site or/and email to the concerned faculty members
End
Algorithm EMPA:
    1. If date=date to send a message to faculty members to offer course(s)
        i. Prepare a message for offering course(s)
        ii. Call IPA to send it to all concerned faculty members
      End If
    2. While (course(s) offering = under process or delay!=null or clash!= null) Begin
        2.1. If clash=normal
                i. EMPA compiles a message as per predefined rules
                ii. Call IPA to send it to all concerned faculty members
            End If
        2.2 If clash=clash caused by faculty members having same priority
                i. Compiles a message to have a meeting to resolve the clash
                ii. Call IPA to send it to all concerned faculty members
            End If
        2.3. If clash=severe clash
                i. Compiles a message to refer the case to HoD
                ii. Call IPA to send it to the HoD
            End If
    2.4. If delay=normal
                i. Compiles a reminder message
                ii. Call IPA to send it to the HoD
        End If
    2.5. If delay = severe
                i. Compiles a message to refer the case to HoD
                ii. Call IPA to send it to the HoD
        End If
    End While
    3. Get all required data from database
    4. Give it a shape of time table
    5. Call IPA to send it to the HoD for verification
    6. If verification = true
            Call IPA to publish the timetable on the web site/send it to all concerned faculty members
    7. Else
          i. Prepare a message as per instructions of HoD
          ii.Call IPA to publish the message/send it to all concerned faculty members
          iii. Go to step 2
    End If
End
```

III. RESULTS AND DISCUSSIONS

The proposed system, in addition to other several features like managing/defining alerts, course settings etc. provides an interface for faculty members to offer course(s). This interface (Figure 5) provides a list of courses which already have been selected by the faculty members. This is just for a view for the faculty member who is currently offering course(s) so that he/she may not select a course which is in this list. But our system provides an option to the faculty members to offer the course(s) which has already been offered. For example, ‘Algorithm Analysis’ is displayed in both lists i.e. ‘Courses Selected by Other Teachers’ and ‘Course(s) List’ If a faculty member selects a course which has already been selected by other faculty member, the EMPA decides as per defined rules (for example, seniority level, already taught course by a faculty member etc.) to which teacher the course should be allocated or refer the case to HoD (in case of sever clash). Figure 5 also provides a list of course(s) which is a mixture of the course(s) that are yet to be selected and that already have been selected by other faculty members. This list is labeled as ‘Course(s) List’. A teacher selects a course which he/she wants to offer and moves to the list labeled as ‘Selected Course(s)’ by clicking the ‘select’ button. A teacher can select maximum three courses in a semester as shown by figure 5. ‘Unselect’ button is used to move back the selected subject(s) from the ‘Selected Course(s)’ list to the list labeled as ‘Course(s) List’. ’Show’ button below the ‘Selected Course(s)’ list is for to see other details about a course which have already been selected by another teacher. The details may include room, day, time and personal information of the teacher who has selected the course.



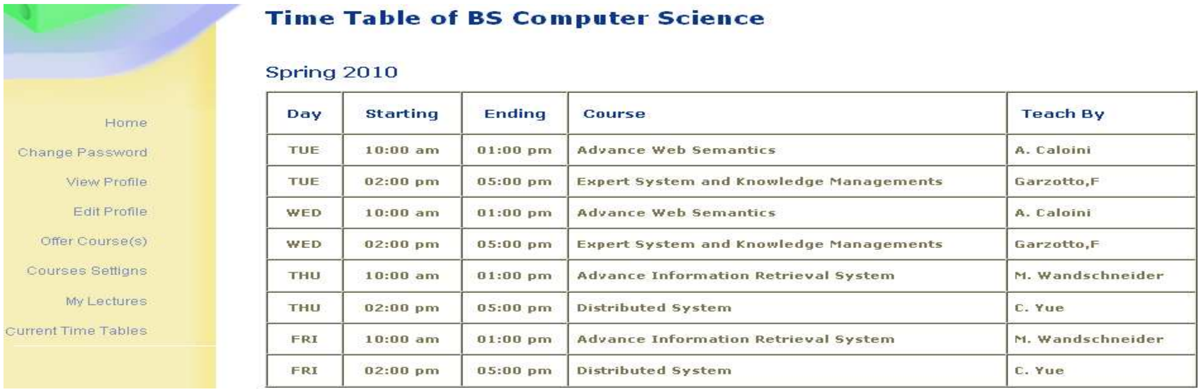
Offer Course(s)

BS Computer Science: Spring 2010

Course(s) selected by Other Teachers	Course(s) List	Selected Course(s)	Selected Room	Selected Day	Course Timing	
Algorithm Analysis Operating System Software Eng Web Semantics Requirement Eng Web Engineering	Algorithm Analysis Computer Architecture Software Architecture Web Semantics Distributed System Intelligent Agent Knowldege Discovery	Software Eng Operating System Web Engineering	3 5 8	Monday Thursday Saturday	01:00 03:30 06:00	PM PM PM

Fig. 5: An Interface for Course(s) Offering

If a teacher selects this button after a selecting a course(s) which already have been selected by the other teacher, the details are shown in a new browser window. Similarly, a teacher can chose the preferred room, day and time. All selected data is saved by ICSA in database. As soon as the data is saved in database the EMPA comes into action. In case of any clash or a delay in offering a course by any faculty member EMPA first tries to resolve the issue on its own. If EMPA is not able to resolve the problem then it informs and let the HoD to resolve the issue. 'Clear' button is used to reset all selected values. 'Ok' button calls ICSA to save data in database. 'Cancel' button is used to revert all activates performed. When all faculty members have finished the task of offering course(s), EMPA after verification from HoD publishes the time table on the web site and sends it to all faculty members as well via IPA. A time table developed by collaboration and cooperation of the three agents looks like figure 6.



Time Table of BS Computer Science

Spring 2010

Day	Starting	Ending	Course	Teach By
TUE	10:00 am	01:00 pm	Advance Web Semantics	A. Caloini
TUE	02:00 pm	05:00 pm	Expert System and Knowledge Managements	Garzotto,F
WED	10:00 am	01:00 pm	Advance Web Semantics	A. Caloini
WED	02:00 pm	05:00 pm	Expert System and Knowledge Managements	Garzotto,F
THU	10:00 am	01:00 pm	Advance Information Retrieval System	M. Wandschneider
THU	02:00 pm	05:00 pm	Distributed System	C. Yue
FRI	10:00 am	01:00 pm	Advance Information Retrieval System	M. Wandschneider
FRI	02:00 pm	05:00 pm	Distributed System	C. Yue

Fig. 6: Time Table Developed by Collaboration of ICSA, EMPA and IPA

IV. CONCLUSION AND FUTURE WORK

A number of efforts have been done to the development of intelligent systems for teaching or examinations' purposes but a very few applications exist for the development of time table. The proposed system is a step forward to this. This system provides interfaces to faculty members and head of department to schedule and manage, view and edit all activities relevant to their courses. Three agents named ICSA, EMPA and IPA work collaboratively and cooperatively to develop the time table. ICSA captures data from a web page and saves it in database. EMPA continuously monitors the environment and in case of a change in the environment, it acts promptly and performs his job accordingly. EMPA is a core part of the proposed system and performs the major job in the development of time table. IPA performs the job of publishing data on the web site and/or presenting it to the faculty members as per instructions of EMPA. Algorithms of the three agents to develop the time table are also presented. As a future work, we are planning to make this system a mobile based system. Functionality of the system would be enhanced so that ICSA can capture data from a mobile device, IPA can send all the information to the mobile phones of the users and EMPA like this system performs the core activities.

ACKNOLEDGEMNET

We are thankful to Higher Education Commission of Pakistan for generous financial support.

REFERENCES

[1] O. Mihaela, O.MAS_UP-UCT: A Multi-Agent System for University Course Timetable Scheduling, International Journal of Computer, Communications & Control, 2:94-102, 2007.

[2] F. Sueychyun, University Course Scheduling System (UCSS): A UML Application with Database and Visual Programming, Journal of Computing Sciences in Colleges, 20:160-169, 2005.

[3] N. Ryan, Getting Started With PHP and Smarty Crossroads, 14:9-16, 2008.

[4] C. Nikolaj, On some drawbacks of the PHP platform In Proc. of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, 2008.

[5] B. Antonio, C. Matt, B. Cao, Adding subqueries to MySQL, what does it take to have a decision-support engine? In Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP,pages.49-56, 2002.

[6] Bouvier. Versions and standards of HTML ACM SIGAPP Applied Computing Review, 3:9-15, 1995.

[7] H.W. Yue, Characterizing Insecure JavaScript Practices on the Web In Proc. 18th International Conference on World Wide Web, Madrid, pages 961-970, 2009.

[8] T.Takeshi and T. Megumi. Production of the time table management system of commercial course university In Proc. World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education, 30, pages 48-3052, 2006.

[9] S.F.Ho, S. Deris, H. Mohd and Z. Siti, University Course Timetable Planning Using Hybrid Particle Swarm Optimization In Proc. of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, Shanghai, China, pages 239-246, 2009.

[10] F. Tim, F. Richard, M. Don and M. Robin, KQML as an agent communication language In Proc. of the Third International Conference on Information and Knowledge Management, pages 456–463, 1994.

[11] S. Mary, The Coming-of-Age of Software Architecture Research In Proc. of the 23rd International Conference on Software Engineering, pages 656, 2001.

[12] M. Nenad, S.R. David, F.R. David and E.R. Jason. Modeling software architectures in the Unified Modeling Language ACM Transactions on Software Engineering and Methodology (TOSEM), 11: 2-57, 2002.