Design of Multi-Agent System for Solution of the School Timetabling Problem

César Covantes, René Rodríguez

Universidad Autónoma de Sinaloa,
Facultad de Informática Culiacán, Culiacán, Sinaloa,
Mexico

c.covantes11@info.uas.edu.mx, rene.rodriguez@info.uas.edu.mx

Abstract. This paper presents the analysis and design of a system based on *multi-agent systems* (MAS) by negotiation with JADE framework (Java Agent DEvelopment) to solve the school *timetabling problem.* In the design, the system considers three types of agents; A coordinator agent responsible for instantiate, create and manage the group agents, where the number of teacher and group agents depends on each case study. The group agents perform the negotiation in order to solve the conflicts between all the teacher agents. The system takes the time, space, activities and other type of constraints by FET (Free Timetabling Software) format in an XML and to prove the algorithm were considered for the analysis and experimentation the case studies Belize, Brazil, Spain and UK.

Keywords: Agent, multi-agent system, timetabling, xml, objective function, FIPA.

1 Introduction

Humans on daily basis plan what activities do in a day and in a certain period of time, but the problem is to select, assign resources and time to obtain a set of activities in an organized manner, resulting a schedule where the order and completion time is important. The persons who performs the schedules should consider different factors such as priorities, time to devoted activities, space availability, cost and valuation of the consequences, satisfying a set of hard and soft constraints [19]; the hard constraints are actions that must satisfy all circumstances, while soft constraints represent a greater flexibility, can be able to satisfy or not, reflecting a temporal relation between activities, given the limited capacity of shared resources.

A common problem behind these assignments is the problem of *timetabling problem (TTP).* In the area of computer science, the *timetabling* represent an optimization problem that belongs to the family of the NP (non-deterministic polynomial time) problems [19]. The NP problems have a computational complexity with a large space search or combine all possible solutions to a problem,

pp. 79–95; rec. 2015-08-14; acc. 2015-09-17

where the goal is to find "good" solutions by an evaluation function that describes its quality in an "acceptable" time.

In some cases, the problem is formulated as a *search problem*, trying to find a schedule satisfying all the restrictions (hard and soft), while in other cases, the problem is formulated as an *optimization problem*, trying to find a schedule that satisfies all the constraints hard and minimize (or maximise as appropriate) through an objective function the soft constraint, applying optimization techniques to a search problem.

This type of problem is only permitted only for a small number of cases (*e.g.*, less than 10 courses) [19], whereas real instances usually may involve a few hundreds of courses. The problem is still present, even though there are different methods that have been developed and used to solve the *timetabling problem* on specifics departments and institutions which are not universal methods so a proposed solution cannot solve "any TTP" problem [20].

The different proposed methods to solve the timetabling are rarely compared with each other by the lot of number of different variables and different ways of quantifying the constraints raised from different policies and practices, which each has its particular characteristics courses assignment. The comparison is necessary to determine what is or are the best computational methods given the different types of data schedules, allowing discard simple techniques, through a manner in which the information it is represented and exchange and unify the restrictions given the different institutions. The eXtensible Markup Language (XML) is used as a standard for data storage, making it useful for several applications that communicate with each other, in addition to exchanging information between different platforms.

2 Theoretical Framework

2.1 School Timetabling Problem

In the school timetabling problem there are "participants" in a fairly general sense, *i.e.* teachers, classes, lecture halls, laboratories, pieces of equipment, and so on. In addition, there exists a set of "hours", sometimes called time slots or periods. The term "availabilities" describe for every participant the subset of hours in which he (it) is free, willing or able to participate in one of the lessons, lectures, conferences or examinations in which he (it) is involved. The latter events are subsumed under the notion of a meet. Every "meet" is described by the collection of participants which have to come together and by the number of hours required for it [20]. The class-teacher timetable problem is obtained if every meet keeps busy exactly one teacher and one class as participants. However, there may be a demand for a meet, which consists of a gymnastics lesson to be held by a male teacher and a female teacher each in different gymnastic hall at the same time. Finally, there may be preassignments of some meets to hours.

Given such a situation, a timetable is a schedule assigning to all these meets the precise number of hours required, so these hours are available for all participants of the meets and such that, as a fundamental requirement, none of the participants is scheduled twice in the same hour.

However, there is a diversity of special requirements a timetable must observe depending heavily on the type of school and on administrative peculiarities of the country [20]. If not, can be problems, is why the problem should consider the following recommendations: consider the division of the set of hours into days if the scheduling cycle is the week. For some participants (namely the classes), it is necessary to avoid free hours between other lessons; they may have free hours only at the beginning or the end of a day. Some subjects require consecutive hours not straddled by a break. There may be limitations on a teacher's daily load, and it may be necessary to provide every teacher with a free day. Subjects taught several times a week should be spread evenly throughout the week. Teachers may indicate a preference on the length of the interval between their lessons. Of course, not all of their claims are equally important; some are merely aesthetic constraints (soft).

The requirements between school levels are different. While in schools, the size of a class is of minor interest, it becomes important at universities, because the number of students in a lecture may vary [20]. In universities the rooms can be selected from a set of rooms of comparable size, while schools each class can be busy all the time by the same teacher. On the other hand, the requirement for the distribution of free hours over the week of either students of lectures are far less restrictive.

There are a large number of variants of the timetabling problem, which differ from each other on the type of institution involved (university or school) and the type of constraint. Therefore [19] classify the timetabling problem into three main classes:

- School timetabling,
- Course timetabling,
- Examination timetabling.

In *school timetabling* the scheduling is weekly for all the classes at school, avoiding two teachers meeting classes at the same time, and vice versa, while the *course timetabling*, the scheduling is weekly for all the lectures of a set of university courses, minimizing the overlaps of lectures of courses with common students. The *examination timetabling*, the exams scheduling is for a set of university courses, avoiding the exams course overlapping with common students, and spreading the exams for the students as much as possible. Such classification is not strict, some problems can fall between two classes, and cannot be easily placed within the above classification [19].

2.2 Distributed Approaches

This section provides a review of work with MAS, in the literature there are a large set of optimization techniques to solve the TTP from the appearance of meta-heuristics in 1983 [21]. In [7], proposed a resolution with multi-agent once the schedule is built, presenting problems of allocating the groups when they are moving from one room to another. A year later[5] arise a model to decompose the secondary scheduling problem into sub-problems and solve each sub-problem in parallel by a decomposition algorithm used to divide a graph in sub-graphs running on a different machines.

After that, in [25, 26] presents two works with MAS. The first, used two mobile agents whose behavior was to verify daily conflicts, while the second work, the agents represent the hard constraint. In [11] presents a hierarchical approach combining a small recursively form in a large one. Also, in [12] presents the problem of school TTP using the model of parallel processing using 'coarse grained' facilitating the exchange of cases and the creation of multiple solutions in parallel. Finally, [15] present the problem *course timetabling* with MAS from a distributed approach.

In all works with MAS, are limited in the specifications of the restrictions and the way which the agents resolve the conflicts, also the works doesn't have a good abstraction and implementation of the problem lacking of information to be implemented in other case studies, being then particular implementations.

2.3 Problem Representation

The unavailability interchangeable TTP benchmarks in a uniform format to express different sets of data and facilitate the public use of these problems, until in [18] UniLang language is presented to define school, course, examination in a language similar to XML format, presenting a tester to validate if a scheduling satisfy with all the requirements and limitations defined by the problem.

In [10] is created the language called STTP (School TimeTabling Problem) with the same XML structure to specify the TTP for high schools and evaluate solutions to these problems. Then, in [17] takes a similar approach to XML format with his own specifications and structure called XHSTT (Xml High School TimeTabling) to facilitate the exchange of data and promote the research in this area for high school with 16 different case studies of different countries located in a repository for a public access with an evaluator.

In [4] an XML format is presented with his own structure and extension called FET with 22 case studies of different countries, located in a repository for a free access. FET has specifically different types of restrictions to their representation in addition used as a tool to capture the problem.

2.4 Existing Software

Nowadays exist software commencial and free that tries to solve the TTP. Every software uses their own optimization techniques and structure restrictions, while others only are a support tool for the allocation of schedules manually. Within the commercial software are aSCTimeTables and Mimosa; aSCTimeTables [1] allows the automatic scheduling which does not specify the search method and the restrictions. The Mimosa software [14] allows the automatic scheduling for schools, meetings, courses and conferences planning without specifying the solution method and restrictions.

Moreover in free software are FET, iMagic, Mimosa 12 Scheduling Software, openSIS, TimeFinder, Lantiv and University Timetabling Comprehensive Academic Timetabling Solution; FET [4] is a GNU license which allows the scheduling automatic in particle swarm, specifying also its structure of the problem and restriction in XML format. iMagic [8] allows the TTP automatically without specifying the search method for a solution and a restrictions. Mimosa 12 Scheduling Software [14], has also a free version with a solution automatically without specifying the solution method and format restrictions. openSIS [16] allows the TTP, also as a tool to fill the grades and inscriptions information without specifying the solution method and format restrictions. TimeFinder [22] uses an optimization algorithm which is not specified as well as the restrictions. Lantiv [13] is a software that serves as a tool for programming schedules manually without specifying the structure of the restrictions. Instead, University Timetabling Comprehensive Academic Timetabling Solution[23] is open source with GNU license that allows the scheduling of examinations and uses minimizing conflicts with local search using constraint programming (variables, values and limitations) without specifying the format restrictions.

3 Multi-Agent System

A Multi-Agent System (MAS) is a system in which several agents interact and pursue a set of goals or tasks to achieve a goal [24]. The agent's features have utility function on a set of goals. Among the techniques that the utility function agents have is to increase it in the points below:

- Coordination,
- Cooperation,
- Directed behavior,
- Planning,
- Communication.

3.1 Agent Platform

Exist an organization called The Foundation for Intelligent Physical Agents (FIPA) responsible to produce software standards specifications for heterogeneous and interacting agents and agent based systems. In the elaboration of those standards, it is used for interoperability between utility-based agent developed by different companies and organizations. The FIPA organization belongs to the IEEE Computer Society standards for their standards, interoperability and software development that promote with the agent technology and other technologies [5].

The agent platform (AP) provides the physical infrastructure in which agents can be deployed. The AP consists in machines(s), operating system, agent support software, FIPA agent management components (DF, AMS and MTS) and agents.

3.2 Agent Management Reference Model

Agent management provides the normative framework within which FIPA agent exists and operates. It establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agent [6].

The agent management considered an AP which provides the physical infrastructure in which agents can be deployed. The AP consists of four components called Agent, Agent Management System (AMS), Directory Facilitator (DF) and Message Transport Service (MTS). The first three components are especial types of agent that support and management of agent, while the MTS provide a delivery message service. The function of this element is located in the specification on web site of FIPA [6].

The internal design of an AP is an issue for agent system developers and is not a subject of standardization within FIPA and the entities contained in the reference model (Fig. 1), between then: external software, agent, agent management system, directory facilitator and message transport service.

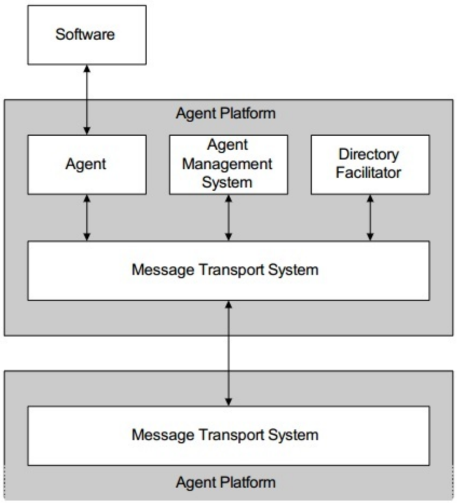


Fig. 1. Reference architecture of a FIPA Agent Platform.

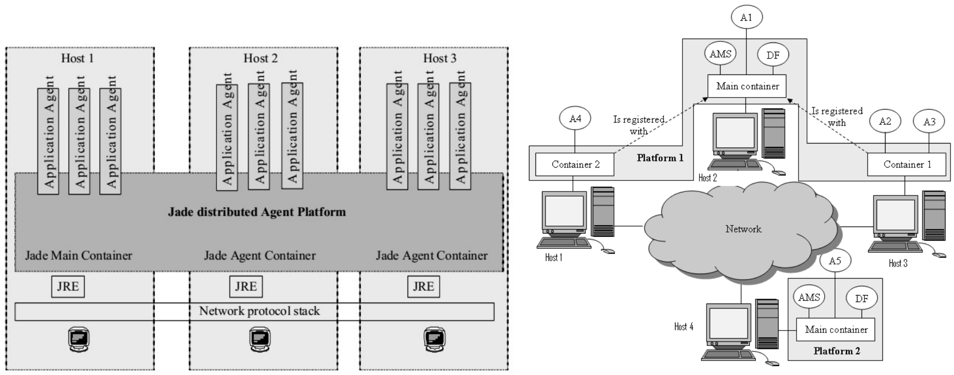
3.3 Agent Platform Implementation

An AP provides the physical infrastructure in which agents can be deployed for archives their goals. FIPA presents a list of mayor implementation of MAS development with public access [5].

Of the 11 platforms that meet the standards of FIPA, JADE was considered because has the advantage of being updated [9] unlike April, FIPA-OS, Grasshopper, Java Agent Services API, LEAD and ZEUS. Also, JADE is free and Open

Source unlike CAPNET and JACK that need license. JADE present security features for authentication connections, user validation and message encryption. Besides, presents a complete graphical interface, great documentation and high acceptance in companies, scientific community and development projects.

In addition [9] has the advantage of being able to distribute into different containers or equipment in a remote mode in order to reduce the number of threads per host on different computers even though they don't have the same operating system. Besides, another feature of JADE is that can be managed through a graphical interface for the communication between platforms. JADE system is made by one or more agent containers, each one with different java virtual machine (Fig. 4).



A) JADE distributed 1.

B) JADE distributed 2.

Fig. 2. JADE Agent Platform distributed over several containers.

4 Methodology

4.1 Constraints Representation

This section describes the structure and design of FET. First FET has the advantages of being available, defined structure, easily public access, many case studies from different countries presented, besides being portable with many constraints divided into four groups shown below:

- Time,
- Space,
- Activities,
- Other type.

The 'time constraints' have the constraint 'not available' that correspond to the day time where teacher or group mustn't be an activity assigned. The constraint 'max days per week' are the days allowed per week that a teacher could teach a class, several teachers, a group or several groups for activities. The 'max gaps per week or per day' are an unused timeslot or several between two activities. The constraints 'max or min hours daily' are the hours or groups can have daily and finally 'max hours continuously' are the hours that can be assigned to a specific teacher or group or all the teachers or groups belonging to the institution, this constraint can be used to affect the minimum number of gaps per day.

In the space constraints 'Home room' is when a teacher, teachers, group or several groups have a default room or rooms. While 'Max building changes per day or week' are the times they have permitted to move between buildings per day o per week and finally the restriction 'Min gaps between building changes' are an unused timeslot or several between two building changes.

In order to know the activities restriction it is necessary to know how an activity is made. An activity are made by a teacher, subject, students (group), duration, total duration, an identifier (id), and identifier of group activity, active and comment, which a teacher has a set of activities that are linked with the corresponding group, so a set of activities make up a schedule.

Given the attributes of an activity, it's possible to assign them either individual or in a group, within the 'activities constraints', the constraint 'An activity has a preferred starting time' is when one activity have a special period to be assigned, moreover the constraint 'An activity has a set of preferred starting time' correspond the set of activities preferred to be assigned. The slot constraint have 'An activity has a set of preferred time slots' is when one activity has a set preferred slot to be assigned. The constraint 'A set of activities has a set of preferred time slots' correspond a set of activities. The constraint 'Min n days between a set of activities' is when a set of activities should be instructed on different day. The constraint 'An activity ends students day' is the activity that mush end a student's day and that activity have a common attribute like a special subject, while the constraint 'A set of activities ends students day' correspond a set of activities. The activity 'A set of activities has same starting time' is when a set of activities should be starting in the same day an hour, while the constraint 'A set of activities has same starting day' is only for a day and the constraint 'A set of activities has same starting hour' is only for an hour.

Also the constraint '2 activities consecutive' are activities that are one after each other as opposed to '2 activities are ordered' which the activities can be assigned in the morning and the other in the afternoon no matter other activities is or are in the middle. The constraint 'Min gaps between a set of activities' are the slot unassigned to allow mobility group and finally 'A set of activities are not overlapping' the activities never be assigned in the same slot.

The 'other restrictions' not correspond any of the above classifications but are important, within them are the 'Basic compulsory time constraints', 'Basic compulsory space constraints', 'Break', 'A room is not available' and 'Preferred room(s)'. The constraint 'Basic compulsory time constraints' a teacher never has to instruct two or more activities at the same time, also students must have maximum one activity per period. In the 'Basic compulsory space constraints' the rooms will never include two or more activities. The constraint ‘Break’ is the way to say that all teachers and students are not available regularly to indicate a break for lunch and finally ‘A room is not available’ for a subject, group, teacher or activity.

4.2 Multi-agent System Design

In the system design of multi-agent platform for the solution of *timetabling problem* the agents/actors are created in 3 different stages throughout the system (Fig. 3). The stages of multi-agent system are explained in detail in the following sub-paragraphs.

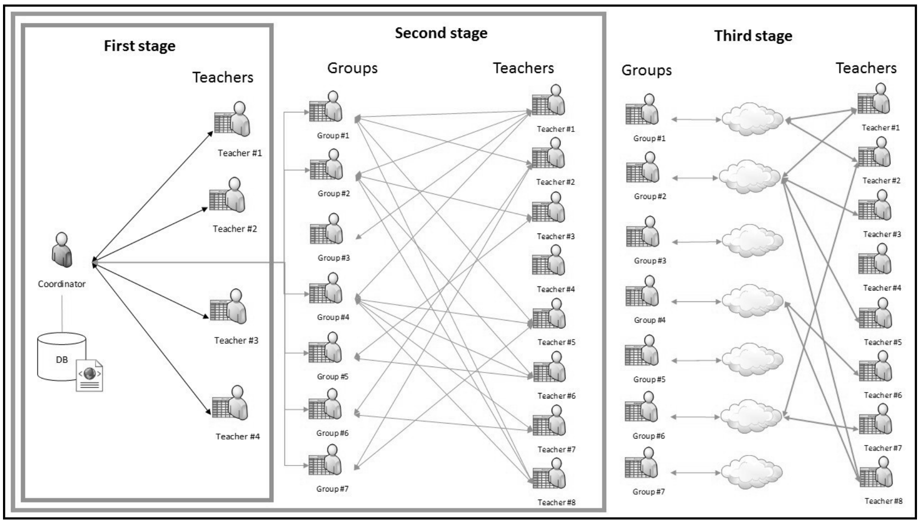


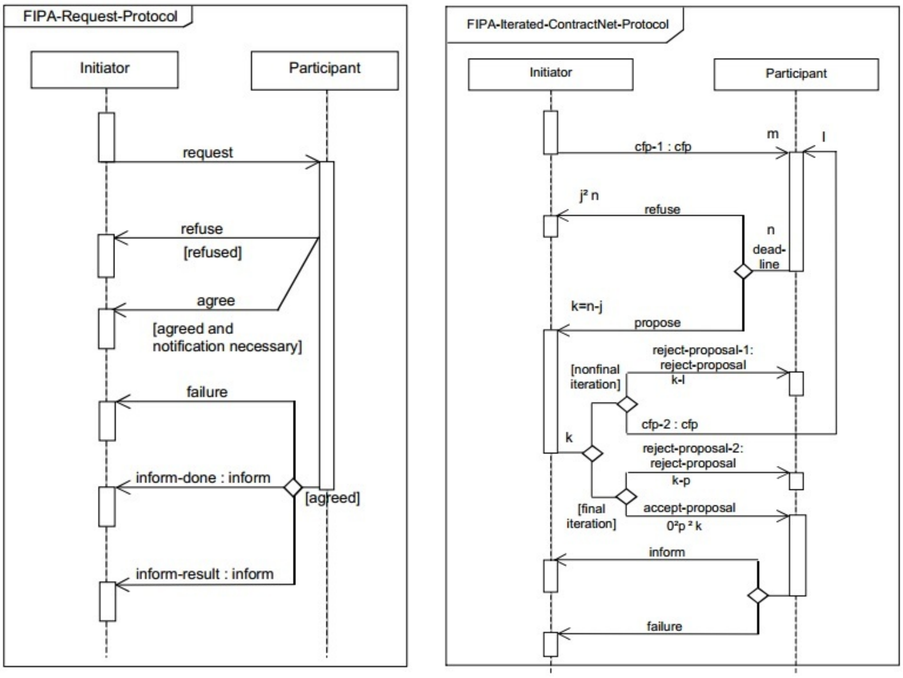
Fig. 3. Design of multi-agent system (MAS).

4.3 Negotiation Strategy Design: First Stage

In this first stage a coordinator agent is created, register with the DF and is the responsible for manage the agents teachers and groups. The coordinator take the activities constraints through an XML file that contain the information of the institution of the case study, also the coordinator agent read the file and create the agents teachers who are registered with the DF, their obtain their constraints and activities, then based on their activities, perform their selfish proposal individually according to their preferences, as long as their respect the general constraints of the institution but respecting part of the schedule suggested by the teacher. The agent professor when finished making the schedule, notify the coordinator.

4.4 Negotiation Strategy Design: Second Stage

In this second stage when all the teachers are in status ready, the coordinator agent creates the agents groups, and each group is register with the DF, the groups obtain the restrictions, activities and the request the schedule of every teacher using the FIPA-request protocol (Fig. 4.A), being the initiator the group on request the teacher schedule and the participant the teacher sending by a response his/her schedule.



A) FIPA request protocol - Second stage. B) Iterated Contract - Net Protocol.

Fig. 4. FIPA Protocols.

4.5 Negotiation Strategy Design: Third Stage

Once all teachers have sent their schedule to the agents group, the agents group verifies if exist a conflict; if exist with a day and hour, the group sends the schedule with the available space to all the teachers involved in the conflict to propose a new position until the agents teachers does not present any problem to move or when they respect the largest number of restriction by an evaluation function. The teachers send the proposal requested by the agent group who evaluates all the proposals received in order to accept/reject the proposals or if is necessary request another position (day and hour) to the agent teacher if the space has already been occupied (assigned) for another negotiation that perform in parallel because exist several negotiations and wins who evaluate and assign first.

The above, for being a distributed approach where all groups are trying to resolve conflicts simultaneously with all teachers, there are cases where a teacher has a conflict with other teachers in several hours with different groups and the same teacher send the same position for different groups, so the group validates and if it was assigned the agent group request a new negotiation to the teacher to provide a new position. The implemented negotiation protocol is the iterated contract-net where de Initiator is the agent group and the Participants the teachers (see Fig. 4.B).

In contrast to contract-net protocol, the iterated contract-net protocol allows new negotiation rounds and is useful when there are cases as mentioned above with several negotiations in occupied spaces, and then the group request each teacher to send another proposal. The way in which the agent group accepts or rejects the proposals of teacher's agents will be explained below with six possible scenarios with three teachers as show in Table 1.

Table 1. Scenarios with three teachers.

Scenario 1	**Scenario 2**	**Scenario 3**
Teacher 1: 0	Teacher 1: 100	Teacher 1: 100
Teacher 2: 0	Teacher 2: 0	Teacher 2: 100
Teacher 3: 0	Teacher 3: 0	Teacher 3: 100
Scenario 4	**Scenario 5**	**Scenario 6**
Teacher 1: 100	Teacher 1: MAX	Teacher 1: MAX
Teacher 2: 200	Teacher 2: MAX	Teacher 2: MAX
Teacher 3: 300	Teacher 3: 300	Teacher 3: MAX

In an evaluation of conflicts that makes a group in the scenario 1, the three teachers no present a problem to move to another space, so the agent group cancel the request for only one teacher randomly, while others accept and change them. In scenario 2 the teacher 1 can be change for a new position but not respecting one or more constraints with a weight of 100 as a result of an evaluation function while the others teachers have no problem to move. The group cancels the request of professor 1 and accept the new position the teachers 2 and 3. In the scenario 3 due all the teachers have problems in enforcing restrictions with the same weight, the group performs the same procedure in the stage 1, cancel one at random and accept the rest teachers.

In scenario 4, the group cancel the request of professor 3 because present the major problem to move, while the others the group accepts them his new position. In stage 5, the teachers 1 and 2 have a maximum value implying that they didn't find an available position, which whereby the group cancels one of the professors 1 and 2 at random and only accepts the teacher 3, and finally on stage 6 none teacher found a new position which the group accepts the teacher request without making changes.

5 Experiments and Results

5.1 Selection of Case Studies

This section describes the experiments and results obtained by implementing the algorithm based in multi-agent system negotiation to resolve conflicts between teachers. In this experimentation process they were considered four case studies at random from a 22 countries. These case studies are freely available in a repository by FET [4].

The countries considered for the experimentation are; Belize, Brazil, Spain and UK, which each one have a specific characteristics in hours, day, groups, teachers, subjects, and activities. The activities of Belize and Spain were modified because there were null data, Belize with a total 952 were 249 null data, while Spain a total of 1086 activities were found 269 data null, giving as a result the data shown in the table 2.

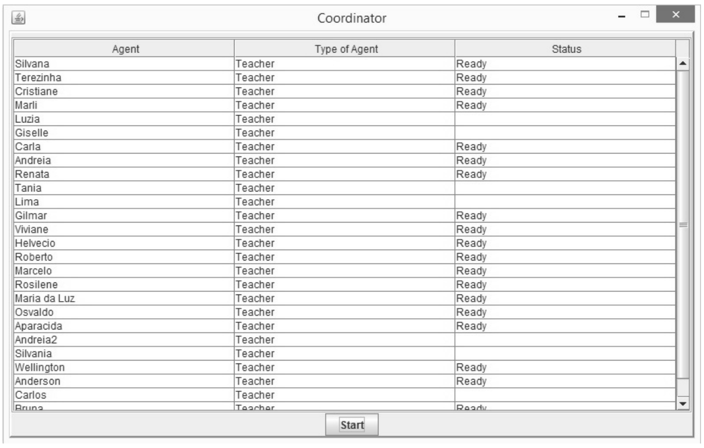
The countries considered for the experimentation are; Belize, Brazil, Spain and UK, which each one have a specific characteristics in hours, day, groups, teachers, subjects, and activities. The activities of Belize and Spain were modified because there were null data, Belize with a total 952 were 249 null data, while Spain a total of 1086 activities were found 269 data null, giving as a result the data shown in the table 2.

Table 2. Characteristics of the case studies.

	Belize	**Brazil**	**Spain**	**UK**
Days	5	5	5	6
Hours	6	5	7	5
Groups	23	16	185	46
Professors	44	27	56	26
Subjects	24	12	78	25
Activities	703	400	817	163

Within results also was develop a system to validate the proposed model, in the Fig. 6 shown the interface execution in the case of Brazil and so for each of the selected cases, in order to see the details on the development (cf. [2]). In this execution is shown the agent coordinator; the first column indicates the name of the agent (agent), the second the type of agent (Type of Agent) in case of teacher or group and finally the agent status (Status). The Fig. 6 refers to the first stage that was shown in Fig. 3.

Also, to validate the second stage of the model in Fig. 3, the system shown in Fig. 6 was implemented once all teachers are ready in status 'Ready', the user



Agent	Type of Agent	Status
Silvana	Teacher	Ready
Terezinha	Teacher	Ready
Cristiane	Teacher	Ready
Marli	Teacher	Ready
Luzia	Teacher	
Giselle	Teacher	
Carla	Teacher	Ready
Andreia	Teacher	Ready
Renata	Teacher	Ready
Tania	Teacher	
Lima	Teacher	
Gilmar	Teacher	Ready
Viviane	Teacher	Ready
Helvecio	Teacher	Ready
Roberto	Teacher	Ready
Marcelo	Teacher	Ready
Rosilene	Teacher	Ready
Maria da Luz	Teacher	Ready
Osvaldo	Teacher	Ready
Aparacida	Teacher	Ready
Andreia2	Teacher	
Silvania	Teacher	
Wellington	Teacher	Ready
Anderson	Teacher	Ready
Carlos	Teacher	
Bruna	Teacher	Ready

Fig. 5. Screenshot coordinator GUI with teachers.

can press the button *Start*. The teachers send the proposals to the groups and the groups check the spaces where exist conflicts to start the part 3 of negotiation between the teachers involved.

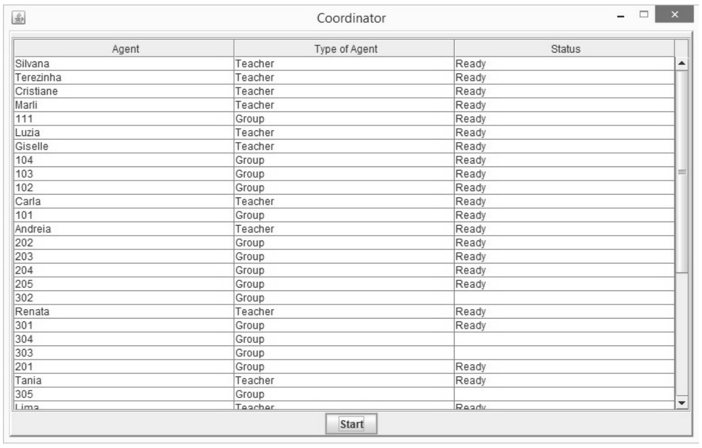
5.2 Conflict Resolution

In the experiment, for each case study were considered 30 executions in order to identify the average of the conflicts presented by each group as well as the unresolved conflict, the initiated protocols and the number of message between agents. The table 3 shows the results of case studies.

Table 3. Results of case studies.

	Found conflicts			Resolved conflicts			Initiated protocols			Messages		
	Total	Average	SD	Total	Average	SD	Total	Average	SD	Total	Average	SD
Belize	3752	125.07	6.02	3505	116.83	5.25	11340	378	8.26	50845	1694.83	44.74
Brazil	4149	138.30	7.30	3067	102.23	5.72	11095	369.83	8.67	51369	1712.30	49.77
Spain	369	12.3	2.83	369	12.3	2.83	13800	460	3.29	42865	1428.83	22.05
UK	3	0.1	0.31	3	0.1	0.31	7083	236.1	0.31	21260	708.67	2.04

The results of Belize the average have 125.07 (± 6.02), are resolved 116.83 (± 5.25) and 8.23 (± 3.18) are unresolved, while the initiated protocols the average is 378 (± 8.26) and messages between agents 1694.83 (± 44.74). In the results of Brazil the average have 138.3 (± 7.3) conflicts found in a initial way, 102.23 (± 5.72) are resolved with the algorithm and not resolve by the algorithm 36.07 (± 6.83), while the average in the initial protocols are 369.83 (± 8.67) and the exchange message between agents are 1712.3 (± 49.77). The results of Spain show

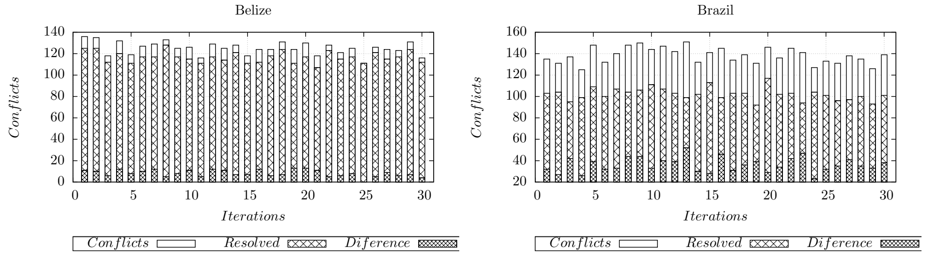


Agent	Type of Agent	Status
Silvana	Teacher	Ready
Terezinha	Teacher	Ready
Cristiane	Teacher	Ready
Marli	Teacher	Ready
111	Group	Ready
Luzia	Teacher	Ready
Giselle	Teacher	Ready
104	Group	Ready
103	Group	Ready
102	Group	Ready
Carla	Teacher	Ready
101	Group	Ready
Andreia	Teacher	Ready
202	Group	Ready
203	Group	Ready
204	Group	Ready
205	Group	Ready
302	Group	
Renata	Teacher	Ready
301	Group	Ready
304	Group	
303	Group	
201	Group	Ready
Tania	Teacher	Ready
305	Group	
Lima	Teacher	Ready

Fig. 6. Screenshot coordinator GUI with teachers and groups.

the found conflicts and resolved with the same average 12.3 (± 2.83), taking a little variability between the conflicts fully resolved with the algorithm, while in the case of UK only 3 iterations showed conflicts which could solve having an average 0.1 (± 0.31).

The following figures show the result of table 3 by iteration according to conflicts found, resolved and could not solve it. In Fig. 7.A the result shown Belize, while in Fig. 7.B the results of Brazil.



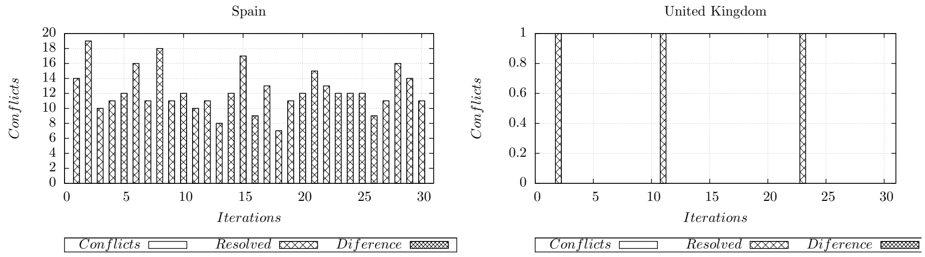
A) Belize resolved conflicts.

B) Brazil resolved conflicts.

Fig. 7. Belize and Brazil resolved conflicts.

The Fig. 8.A shows the results of the case study Spain, which shows the overlapping results from the conflicts because the algorithm could solve in fully the conflicts, while Fig. 8.B shows the results of the case UK with only 3 conflicts that could be resolved with a difference of zero having no conflicts in other iterations.

The graphs show the variability of conflicts for cases of Belize and Brazil, the



A) Spain resolved conflicts.

B) UK resolved conflicts.

Fig. 8. Spain and UK resolved conflicts.

graphs behave similar because its characteristics has less space for the allocation the activities of teachers in contrast to Spain and UK, also in conflicts found in Belize and Brazil the teachers from an initial state have a schedule that more adapt his/her convenience, imposing more restrictions, while in Spain and UK the teachers impose fewer restrictions, resulting in greater variability in assignment activities.

6 Conclusions and Future Work

The contributions in this work is a system that implements a multi-agent system through negotiation to resolve the *school timetabling problem* allowing the incorporation of general constraints to model different case studies, regardless of the institution or country that want to solve as long as comply with the restrictions and standards of FET.

Another contribution of this work is to allow the agents to interact and solve conflicts in a dynamic way and not deterministic, also one of the advantages of the proposal is that it is compatible with FIPA specifications to implement recognized standards, as well as present regardless of the operating system. In the part of the system like advantage to implement multi-agent system it was possible to consider the initial proposals of teachers, be a difficult task for other techniques that have to generate solutions and validating every assignment.

Based on the result shown in tables and graphs, it's clear that multi-agent systems can in great measure resolve the conflicts; solving in the case of Belize 93.41%, also with Brazil 73.92%, whereas Spain 100% and UK 100%.

As a part of the future work it is necessary to implement the missing constraints. Validate the results obtained with Multi-Agent Systems and compare the solutions generated by FET that implements a particle swarm technique.

Another point to be addressed in future research will implement a new technique to solve conflicts that they could not resolved through a similar technique proposed by researchers [3] through Eco-Problem-Solving.

References

1. Consultants, a.A.S.: aSc timetables. http://www.asctimetables.com/timetables_en.html (2013)
2. Covantes, C.: Sistemas Multi-Agentes para la solución del problema de programación de horarios escolares mediante negociación. Master's thesis, Universidad Autónoma de Sinaloa (2014)
3. Drogoul, A., Dubreuil, C.: A distributed approach to n-puzzle solving (1993)
4. FET: FET free timetabling software. http://lalescu.ro/liviu/fet/ (2013)
5. FIPA: The foundation for intelligent physical agents. http://www.fipa.org/index.html (2012)
6. FIPA AMS: FIPA agent management specification. http://www.fipa.org/specs/fipa00023/XC00023H.html (June 2001)
7. Gaspero, L.D., Mizzaro, S., Schaerf, A.: A multiagent architecture for distributed course timetabling. In: Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling. pp. 471–474 (2004)
8. iMagic Software: iMagic timetable master. http://www.imagictimetablesoftware.com/ (2013)
9. JADE: Java agent development framework. http://jade.tilab.com/ (2013)
10. Kingston, J.H.: Modelling timetabling problems with STTL. In: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III. pp. 309–321. PATAT '00, Springer-Verlag, London, UK (2001), http://dl.acm.org/citation.cfm?id=646431.692901
11. Kingston, J.H.: Hierarchical timetable construction. In: Burke, E.K., Rudov, H. (eds.) Practice and Theory of Automated Timetabling VI, Lecture Notes in Computer Science, vol. 3867, pp. 294–307. Springer Berlin Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-77345-0_19
12. Kingston, J.H.: Solving the general high school timetabling problem. In: Proceedings of the 8th international conference on the practice and theory of automated timetabling (PATAT 2010). pp. 517–518 (2010)
13. Lantiv: Lantiv Scheduling Studio. http://schedulingstudio.com/ (2013)
14. Mimosa Software Ltd.: Mimosa softwre. http://www.mimosasoftware.com/ (2013)
15. Obit, J., Landa-Silva, D., Ouelhadj, D., Vun, T.K., Alfred, R.: Designing a multi-agent approach system for distributed course timetabling. In: 11th International Conference on Hybrid Intelligent Systems (HIS). pp. 103–108 (2011)
16. OpenSIS: OPENSIS. http://www.opensis.com/features.php (2013)
17. Post, G., Ahmadi, S., Daskalaki, S., Kingston, J., Kyngas, J., Nurmi, C., Ranson, D.: An XML format for benchmarks in high school timetabling. Annals of Operations Research 194(1), 385–397 (2012), http://dx.doi.org/10.1007/s10479-010-0699-9
18. Reis, L.P., Oliveira, E.: A language for specifying complete timetabling problems. In: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III. pp. 322–341. PATAT '00, Springer-Verlag, London, UK (2001), http://dl.acm.org/citation.cfm?id=646431.692907
19. Schaerf, A.: A survey of automated timetabling. Artificial Intelligence Review 13(2), 87–127 (1999), http://dx.doi.org/10.1023/A%3A1006576209967
20. Schmidt, G., Ströhlein, T.: Timetable construction an annotated bibliography. The Computer Journal 23(4), 307–316 (1980), http://comjnl.oxfordjournals.org/content/23/4/307.abstract

21. Talbi, E.G.: Metaheuristics: From Design to Implementation. Wiley Publishing (2009)
22. Time Finder: TimeFinder. http://timefinder.sourceforge.net/ (2013)
23. UniTime: University timetabling. http://www.unitime.org/ (2013)
24. Weiss, G. (ed.): Multiagent systems: a modern approach to distributed artificial intelligence. MIT Press, Cambridge, MA, USA (1999)
25. Yang, Y., Paranjape, R., Benedicenti, L.: An examination of mobile agents system evolution in the course scheduling problem. In: Canadian Conference on Electrical and Computer Engineering. vol. 2, pp. 657–660 (2004)
26. Yang, Y., Paranjape, R., Benedicenti, L.: An agent based general solution model for the course timetabling problem. In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. pp. 1430–1432. AAMAS '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1160633.1160901