How to solve a timetabling problem by negotiation *

Marie-Hélène Verrons[1] and Philippe Mathieu[1]

Équipe SMAC, LIFL – CNRS UMR 8022
Cité Scientifique, 59655 VILLENEUVE D'ASCQ, France
{verrons,mathieu}@lifl.fr

1 Introduction

Problems involving resource allocation such as appointment taking or timetable creation have usually been tackled as constraint satisfaction problems. For the two aforementioned applications, this forces the users to reveal their agenda in order to give their constraints. But one doesn't always want to reveal his private events. Thus a solution is for the users to keep their agenda private and to use negotiation to take appointments or courses.

In this article, we present a negotiation based approach for timetable creation. Our aim here is not to present an approach that gives the best results but an original one that is more flexible than others which must restart from scratch at each change in the environment. Using agent based negotiation enables us to dynamically add or remove constraints or agents. We show how to transform the timetabling problem into a negotiation problem between agents. Note that an agent can represent a person as well as a thing such as a room.

To illustrate our purpose, we use a university timetable creation application. A benchmark has been proposed by the group Asa GDR-I3 [1]. This problem needs, in order to provide a solution, to be able to adapt itself in response to dynamic changes in the environment. This problem needs a collective search of the solution, and isn't a simulation problem: the aim isn't to recreate virtually the behaviour of an existing organism, but to furnish an expertise.

In order to develop this application, we use a generic negotiation API: GeNCA [2]. This API provides the whole management of negotiation processes and only needs communication configuration and strategy definition. For the application we want to develop, there'll be no work for communication configuration as we'll use the provided MAS communication (either Magique or Madkit). Default strategies are also provided with GeNCA which are well-suited for our application.

In this paper, we first present the negotiation approach for resource allocation problems and the university timetabling creation application that serves our purpose. Then, we present the negotiation toolkit used to build this application, the way to develop it and the results of this approach.

* This work has received a grant from European Community – FEDER – and "Region Nord-Pas de Calais" – CPER TAC –.

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 502–505. ISBN 80-210-3726-1.

2 How negotiation is helpful

2.1 The University timetable creation problem

The problem here is to create the timetables of students and teachers in a University. We present here the benchmark that has been proposed by the group Asa GDR-I3. Actors (in a UML sense) involved are teachers, student groups and rooms. Each one of these actors (individually) has constraints to be satisfied (at best). A teacher has constraints over his availabilities (day of week, time slot), his skills (particular teaching), and his need of particular teaching equipment such as an overhead projector.

A group of students has to follow a particular teaching composed of a set of several courses of several teaching subjects. For example, x courses of subject 1, y courses of subject 2, and so on.

A room is equipped or not of particular equipments (overhead projector, ...) and can be occupied or not during a time slot of a day.

We assume that for each actor, constraints are given in a list. The order in the list gives the importance of the constraint comparing to the others (the first one can be relaxed easier than the last one). The problem to solve consists of conciliating these constraints in order to propose a timetable for a specified duration.

2.2 Which negotiation system are we using?

Many kinds of negotiation systems exist, such as the Contract-Net Protocol (CNP) [3], auctions, multi-step negotiations or else combined negotiations. The negotiation protocol we use here is an extension of the CNP which adds rounds of counter proposals from the contractors and the manager.

The negotiation system we use allows different actors to negotiate contracts over resources. One actor (the initiator) proposes a contract over several resources to a set of actors (the participants). Each participant answers either by accepting the contract or by rejecting it. If the contract has been accepted by a sufficient number of participants, the initiator confirms it. Otherwise, the initiator asks participants which resources they prefer for the contract. He then chooses another set of resources according to participants preferences and proposes a new contract to the participants. This cycle is done until a solution is found or a predefined number of rounds is reached.

2.3 How to transform a scheduling problem in a negotiation problem

The first thing to do is to determine which are the resources to be negotiated and who are the actors in the negotiation. In the application we use here, resources are naturally time slots. Actors are the teachers, the student groups and the rooms.

Then, you have to define the negotiation protocol that best fits your application. For the timetabling problem, the extension of the CNP presented above is the best one. Initiators are the teachers while student groups and rooms are participants. You also specify some features of the negotiation as the possibility to retract yourself from a contract previously taken and to renegotiate this contract. These features are typically needed for timetabling problems.

2.4 Advantages of this approach

The advantages of this approach are twofold. On the one hand, the use of negotiation allows users to find a timetable that respect their constraints without having to reveal them to the others. It also enables them to manage themselves their constraints and so they choose which one to relax if needed. On the other hand, the multi-agent approach facilitates dynamic changes such as the arrival (or removal) of an actor (agent) and negotiation facilitates the changes of constraints. These dynamic changes are taken into account in real time and don't affect the whole process of finding a solution. That is to say that the process has not to be restarted from the beginning but adapts itself to the changes. The resulting system is thus more flexible facing dynamic changes during the resolution process.

3 Coding the problem with GeNCA

To solve this timetable problem, we propose to use negotiating agents in a multi-agent system. We use the MAS platform Madkit and the GeNCA API which provides a framework for building a negotiation application.

The resources that will be negotiated are the time slots. To solve this problem, we decide to assign an agent to each actor. Assume there are 3 teachers and 3 student groups. Thus, 6 agents are defined: $t1$,$t2$, $t3$, $g1$,$g2$ and $g3$. Agents work on an asynchronous mode. Each teacher inputs his timetable in real time (they use GeNCA as a negotiation help tool), student groups are in an automatic mode, that is to say the agents work as background tasks. Users give priority to resources and to the other actors.

As GeNCA doesn't force to take the contract, it is possible that the whole courses are not scheduled. So with no other features in the agent, if it doesn't succeed in its negotiation, it won't try to move another contract in order to be able to take the one he failed to take. The teacher has to monitor its agent to check that all courses have been taken and otherwise he must cancel courses and move them in order to take the missing ones.

We introduce in the system 3 new agents $r1$, $r2$ and $r3$ that represent the rooms. Now, teachers have to choose the group and the room when they create their contracts. A teacher can select all rooms when he creates the contract so that he can see if there's a place for his course, choose the room he wants if several are free and search another time slot if no room is free. Selecting all rooms isn't a problem as you can choose to confirm the contract for some of the

participants and cancel it for the others. To do so, initiator strategy might be slightly modified in order to keep only one room participant.

Relaxing constraints and adding new ones are already provided in GeNCA, as you can cancel previous contracts and add new ones. If a contract for a course is cancelled, it will be automatically renegotiated.

Having a multi-agent system enables us to add or remove agent in real time, without perturbing the whole application and having to restart from scratch.

3.1 Concrete results

In our experiments, we have used agents that negotiate contracts that have been created by the user. We haven't added to the agent skills to store a list of courses that must be scheduled and to check that they have been scheduled. As GeNCA doesn't specify that the contract must be taken at the end, it is possible that all courses the user wanted aren't scheduled. The strategy used looks at each possibility of free resources (or resources taken by a less prior initiator) before cancelling the contract but it doesn't cancel another contract in order to take the new one. Each teacher must then check that he has all his courses. To face this problem, we should use an agent having those skills. On the contrary, GeNCA is adapted to dynamic changes: if a course is cancelled, it is automatically renegotiated.

4 Conclusion

In this paper we have presented a novel approach to solve a timetable creation problem. Using negotiation to establish timetables is a new and promising research field. It enables teachers to find a timetable taking into account their constraints without giving them to the others. Moreover, it is a flexible approach that facilitates the integration of dynamic changes such as modification of constraints or arrival of an actor. It is important to keep in mind that this approach is interesting because of its flexibility and not for the results it gives. For the moment, we don't try to give the best results for timetabling problems but we want to show that negotiation is an approach that is worth to be explored. More experiments are needed for the timetable creation problem in order automate the whole process. We are at the moment designing more specialised agents that will have the list of courses they must schedule in order to make experiments without human intervention for contract creation.

References

1. ASA group members. Emploi du temps. Technical report, groupe ASA du GDR-I3, 2003. http://www-poleia.lip6.fr/∼guessoum/asa.html.
2. Philippe Mathieu and Marie-Hélène Verrons. A General Negotiation Model using XML. *Artificial Intelligence and Simulation of Behaviour Journal (AISBJ)*, 1(6):523–542, August 2005.
3. R. G. Smith. The Contract Net Protocol : high-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, C-29(12):1104–1113, December 1980.