Timetable Planning using Intelligent Agents

Irina Tudor$^1$, Mădălina Cărbureanu$^1$

(1) Department of Informatics, Petroleum-Gas University of Ploieşti, 39, Bd. Bucureşti, 100680, ROMÂNIA
E-mail: tirinelle@yahoo.com

Abstract

At the beginning of each semester a frequent problem appears regarding the timetable planning. It is not an easy task for the person responsible with solving this problem because he/she must take into consideration many constrains such as: teaching stuff timetable options, elaboration rules (number of lectures, number of seminars, number of practical activities per day, etc.) and the number of students which determines the allocated lecture/seminar hall. In this case, an adequate method for reducing the execution effort and time is the designing a multi-agent system (MAS) using ZEUS software, developed by British Telecom. Our work consists in identifying and implementing the necessary agents with their tasks for sharing the resources, establishing the communication ontology and coordination. This paper highlights the opportunity of multi-agent systems application in the superior education field.

Keywords: Intelligent Agent, Ontology, Multi-Agent System, Education.

1. Introduction

In recent years, a major research effort in artificial intelligence domain has been invested in designing and building intelligent agents-software or hardware entities that interact with an external environment in an intelligent way.

In computer science, an intelligent agent (IA) is a software agent that assists users and acts on their behalf, in performing non-repetitive computer-related tasks, in the sense of a representative agent. Intelligent agents are used for operator assistance or data mining. The intelligence implies the ability to adapt and learn (Wikipedia, Software agents, 2008).

In artificial intelligence, an agent is used for intelligent actors that observe and act upon an environment, in the sense of a rational agent.

In our paper we present an application of intelligent agents in the superior education field. A solution for timetable planning in the framework of a department is given.

2. About Intelligent Agents

In literature, an agent is known as an entity that perceives, reasons and acts. In computational terms, that which is perceived is an input, to reason is to compute, to act is to output the result of computation. An agent is equipped with objectives and the rational quality consists in acting optimally with respect to its objectives.
Intelligent agents perform a wide range of services, including automatic searching, answering specific questions, providing information and updates about events, running programs and presentations, reporting current news, comparison shopping, and tutoring (Baylor, 1999). This technology combines artificial intelligence (reasoning, planning, natural language processing, etc.) and system development techniques (object-oriented programming, scripting languages, human-machine interface, distributed processing, etc.).

An intelligent agent can be used to perform various activities such as: searching for information automatically, answering to the specific questions, informing users when an event has occurred, providing custom news to user on a just-in-time format, intelligent tutoring, automatic services, such as checking web pages for changes or broken links.

An intelligent agent can be applied successfully in various fields, as follow: workflow management, network management, air-traffic control, business process engineering, command and control, education, digital libraries, information management, data mining, electronic commerce.

An agent runs if two conditions are met. The former is a common language, called Agent Communication Language (ACL), that must exist in order to enable software to recognize the intention behind a request of an agent and, as a latter condition, there must be an architecture, where a piece of software can describe its abilities and needs.

Various special languages have been developed to facilitate the communication between agents and the most common are: FIFA ACL, Actor, Tcl/Tk, Telescript, Linda (mostly for mobile agents), Agent0, Concurrent Metaterm, KQML, etc. Communication protocol is not a low-level protocol but a protocol establishing possible actions of agents in every moment of communication with other entities.

A FIPA ACL message contains a set of one or more message parameters. Precisely which parameters are needed for effective agent communication will vary according to the situation. The only parameter that is mandatory in all ACL messages is the performative one, although it is expected that most ACL messages will also contain sender, receiver and content parameters. If an agent does not recognize or is unable to process one or more of the parameters or parameter values, it can reply with the appropriate not-understood message (FIPA Abstract Architecture Specification. Foundation for Intelligent Physical Agents, 2000).

3. Multi-Agent System Design

In this application there are defined four agents, capable to interact in order to minimize the effort and time for timetable planning process.

The above-mentioned agents are the following: Timetable administrator, Timetable teacher, Environment and Restriction. Also, one may use utility agents that are automatically generated by Zeus: Broker, Visualiser and ANS (Agent Name Server).

Various roles and tasks are associated to each agent. For instance, the functions for Timetable_admin agent are:
1. Requesting the teachers their timetable options;
2. A new timetable option solicitation, when the current option is not valid;
3. The timetable supplying;
4. The supplying of the laboratories loading.

For the Timetable_teacher agent, the associated functions are:
1. Verifying the laboratory’s state;
2. Delivering laboratory activities;
3. Sending the solicited timetable option;
4. Sending a new timetable option.
The role of the Environment agent is the laboratory information supplying and the only role of the Restriction agent is to assure restrictions application.

For having a good image upon the information flux, the representation of the information flux and interaction diagram is necessary (figure 1, figure 2).

**Figure 1. Information flux**

**Figure 2. Interaction diagram**
An agent PAGE (Perception, Actions, Goals and Environment) description consists in the perceptions, actions, goals and environment (in which the agent interacts with other agent/agents) identification. The table below displays the PAGE description for the proposed agents in our application:

<table>
<thead>
<tr>
<th>Agent</th>
<th>Perception</th>
<th>Action</th>
<th>Goal</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timetable_admin</td>
<td>Receiving options</td>
<td>Asking options</td>
<td>Generating timetable</td>
<td>Laboratory</td>
</tr>
<tr>
<td>Timetable_Teacher</td>
<td>Receiving tasks</td>
<td>Answering tasks</td>
<td>Timetable establishing</td>
<td>Laboratory</td>
</tr>
<tr>
<td>Environment</td>
<td>Laboratory state</td>
<td>Supplying laboratory information</td>
<td>Supplying laboratory information</td>
<td>Laboratory</td>
</tr>
<tr>
<td>Restriction</td>
<td>–</td>
<td>Restrictions applying</td>
<td>Restrictions discharging</td>
<td>Laboratory</td>
</tr>
</tbody>
</table>

4. The Multi-Agent System Implementation with Zeus Agent Toolkit

An essential step in MAS developing process is the knowledge modelling, consisting in the ontology representation. In the context of computer and information sciences, ontology defines a set of representational primitives with which one can model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members).

The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application (Gruber, 2008).

In our application the agent’s ontology is composed from the terms used in the communication process. The figure below presents the developed ontology:

![Agent ontology](image)

*Figure 3. Agent ontology*
The next step consists in the agent’s development and identification of associated tasks. The development of agents for our application is presented in the figure below:

**Figure 4. The agents in Zeus**

The development of an agent, for instance *Timetable_admin*, implies the realization of the next intermediary steps: the Zeus implementation of the agent tasks, establishing of the relation of the agent with other agents (peer to peer, subordinate or superiority relation), establishing the coordination protocols (for the respondent or initiator agent role), allocating the initial resources and establishing the maximum number of simultaneous tasks. All these are established in the following panels: *Agent Definition Panel, Agent Organization Panel, Agent Coordination Panel* and *Value Restriction Panel*.

The implementation of the *Supply_timetable* task associated with *Timetable_admin* agent is presented in the next figure:
Zeus software provides a batch of utility agents: **Agent Name Server (ANS)**, **Facilitator** and **Visualiser**. These agents are implicitly generated by Zeus and they form, beside the user-developed agents, the application agent society. The resulted agents society is presented in the figure below:

**Figure 5. Task Preconditions and Effects**

Using the proposed ontology and the **ACL** as a communication language (Agent Communication Language) for reaching the application goal, the agents start to change messages (i.e., sending call for proposals (cfp), sending proposals, proposals’ acceptance/refusal, sending information, etc.), as presented in the figure:

**Figure 6. Agents society**
As a result of agents communication a timetable is generated in a primary form represented by attributes with values which can be interpreted to obtain a useful timetable form.

*Figure 7. Interactions between agents*

*Figure 8. The results table*
REFERENCES


