SOCRATES AGENTS – IN SUPPORT OF STUDENT MOBILITY

Maria GANZHA, Wojciech KURANOWSKI, Marcin PAPRZYCKI

ABSTRACT: Software agents are often claimed to become a new generation of tools facilitating efficient management of information. While a number of possible agent application areas are listed in the literature, support for academic mobility is not one of them. At the same time student mobility is one of the important objectives within the European Union and autonomous agents could be used to streamline the process and help students that are interested in participating in it. In this paper we discuss an agent system that supports this purpose.

Introduction

One of the more important goals that the European Union is striving at achieving is social mobility and one of interesting aspects of this goal is "academic mobility" understood as mobility of students and faculty. Mobility of "academicians" is supported financially through Marie Curie Mobility Programs. Among them, programs like Socrates and Mundus are designed, among others, to allow students to visit universities in other EU countries and spend there one or two semesters, while obtaining stipend from the EU. Such a visit is possible when: (a) universities have a bilateral agreement and (b) student wins a competition if there are more interested students than the agreed number of exchanges. Note that faculty members can be also a part of Socrates/Mundus agreements and therefore results presented here can be easily extended into support of faculty mobility.

Obviously, arranging a student visit involves a number of administrative steps (further steps are also required post completion of an exchange). Fulfillment of all necessary requirements is a tedious task and takes a lot of energy on the part of the student and resources on the part of the University. Our work, involves creation of an agent system that would facilitate and support a SOCRATES-type mobility program. Work presented here is an extension of results reported in [1].

We proceed as follows. In the next section we summarize steps that have to be undertaken by a student who would like to participate in a mobility program. We follow with the description of the design of an agent system and a few details of its implementation (Section 2). In Section 3, we illustrate the performance of the system by describing experiment involving three countries. We complete this note with a brief description of our future research directions (Section 4).

1. Student mobility – administrative perspective

Let us consider EU-based institutions of higher learning that are to be involved in an international student exchange program. Even in countries like Poland, we can observe increasing role of electronically stored and processed data,

e.g. student records. Furthermore, students, faculty members and administrators communicate using e-mail (to a greater or lesser extent). Finally, some universities provide an interface that allows students to check items like: schedule of selected courses, upcoming exams, earned credits etc. Within this context, let us conceptualize situation when a student from an EU-located university wishes to participate in a student exchange program. In this case the following steps have to be completed (see also Figure 1):

before departure:

(1) Selecting foreign university
(2) Realizing its requirements
(3) Realizing all requirements of students home university
(4) Delivering all necessary data both to the home and to the foreign university
(5) Organizing a place to live at the foreign site

after arrival at foreign university:

(6) Contacting local coordinator
(7) Arranging the schedule of courses
(8) Managing courses and credits required to meet the exchange program agreements

after returning to the home-university:

(9) Completing a survey or delivering a report to the home-site coordinator.

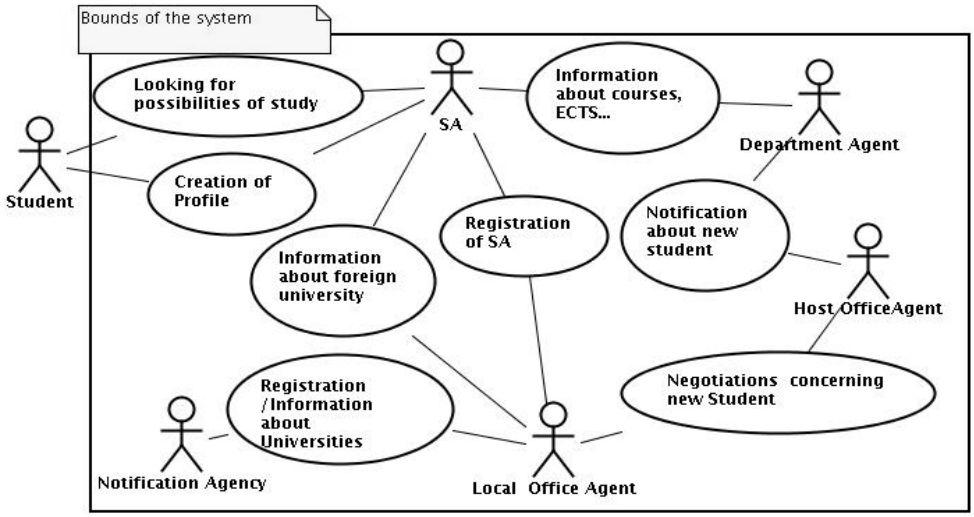

Figure 1. Functional requirements of the system

The first four steps involve mostly interactions between the student and Dean's Office at her local university, as well as an information exchange with the local program coordinator. Let us note here, that in the above description we are silently omitting the case when multiple students are interested in a limited number of openings within the exchange program. This situation is implicitly a part of (1) above, as being selected locally is a part of selection of a foreign university –

student cannot select a university that "does not accept him" (see also depiction of Local Office Agent functionality presented in Figure 3). Step five is often completed "automatically" by an office at the host institution that receives information about incoming exchange students as a part of the document circulation involved in steps 1–4. Otherwise, student has to search a flat or to communicate with a separate organization which supervises dormitories/apartment rental. After arriving at the chosen university student has to contact the local exchange coordinator to arrange the course schedule in such a way to fulfil the requirements of the program (e.g. to accumulate a required number of credits, to study subject areas that were covered by the bilateral agreement etc.).

In all universities, appropriately prepared to handle exchange students, steps 1–4, 6–8 or 9 don't present problems (even if they are not fully supported by electronic means of communication) when considered independently. Problems materialize when all steps have to be completed "together" and thus, when various documents have to circulate (bi-directionally) between different units within university; between different units in different (foreign) universities and, finally, between these units (both local and foreign) and the student. Moreover, since not every university supports electronic data management to the same extent (and some universities in countries like Poland or Romania, have only a very minimal IT support in administration), it is often the case that an extremely large number of documents have to be transferred "manually." This involves sending letters, faxes, receipts (in case of organizing a flat) and/or numerous telephone calls.

2. Student mobility – agent system

The main idea of our project is to develop a solution which would make formalities of taking part in a student exchange program simpler, and also reduce number of issues that presently have to be dealt with "face-to-face." We propose a system that would facilitate the flow of information required to establish participation in an exchange program. Furthermore, as the system develops, it could remove humans from the process (other than the student expressing a desire to participate in it). Let us start from summarizing (in Figure 1) the proposed flow of activities. Here, we have divided the functionalities into the following agents:

Student Agent (*SA*) is the interface between the student and the system and is also students' representative. It can organize or provide view of students' schedule, check the total number of credits acquired thus far, make an appointment with a professor and/or advisor etc. The *SA*, will represent student in organizing participation in the student-exchange program. After student arrives at the foreign university, the *SA* communicates with the *Host Office* and the *Department Agent* and supplies the exchange student with all required information. In Figure 2 we depict the UML state diagram of this agent.

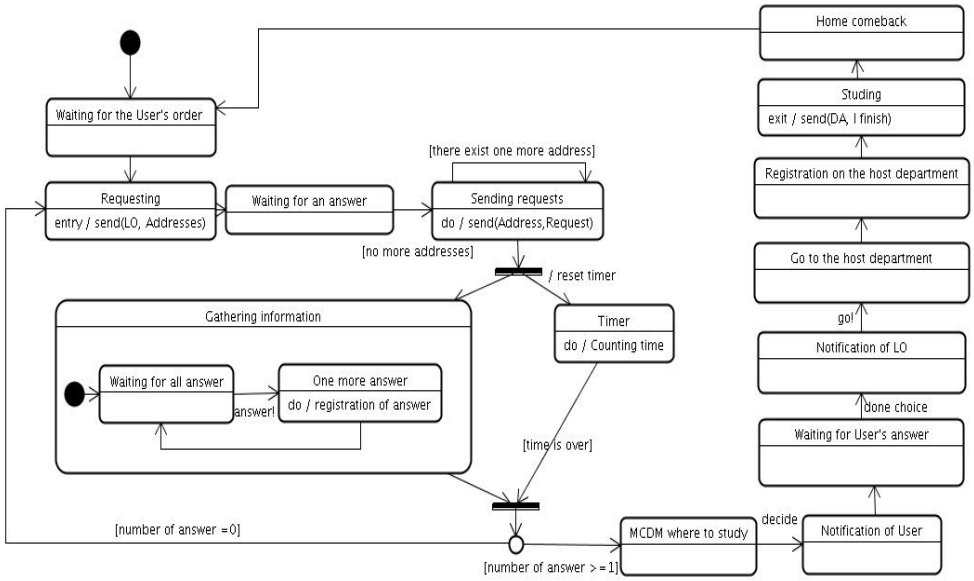

Figure 2. Student Agent State diagram

Local Office Agent (*LOA*) is, among others, a co-ordinator of the Socrates program. *LOA* possesses information about universities that have bilateral Socrates agreements with its university, exchanges messages required to set up departure of a student to another university. Here the contact data of *LOA* is stored in a Yellow Pages service point, which should be understood as a database of agent addresses [3]. In Figure 3 we present the UML state diagram of this agent.

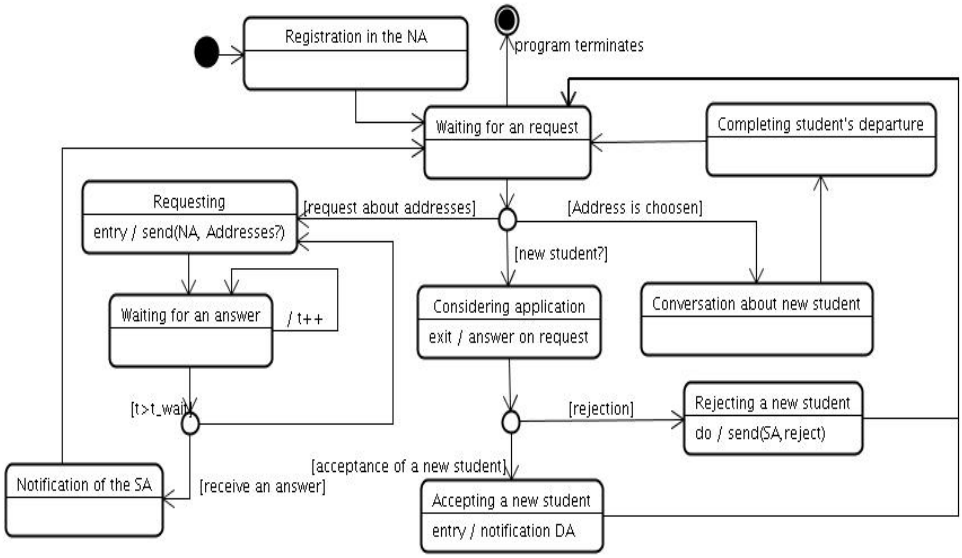

Figure 3. Local Office state diagram

Department Agents (*DA*) may be conceptualized as a combination of virtual department heads and secretaries of individual departments of a particular university. These agents are envisioned to be responsible for courses offered during a given semester, course schedules, and calculation of ECTS and so on.

Host Office Agent (*HOA*) is the *LOA* counterpart at the foreign host institution. In other words, the *HOA* is the *LOA* of the foreign university.

Notification Agency (*NA*) represents the offices in Brussels that have to be notified that a given student will be participating in an exchange program. This notification means (1) that one of the spots in the negotiated bilateral agreement was taken and (2) that a given student will be funded by the Socrates scholarship.

2.1 Agent Interactions

Let us now list interactions between agents that take place when the *SA* attempts at arranging the exchange program for the student (see Figure 1). We assume that the system has been initialized and that the student has communicated with the *SA* and established the selection criteria (e.g. country, subject area, etc.). Then, the system performs the following actions (working autonomously – when student specifies requirements, agents make all decisions). Note that communication between agents is achieved through exchange of ACL messages.

1. *SA* sends search request to the *LOA* to get addresses of all foreign universities that *LOA* has bilateral agreements signed with (in the specified field of study)
2. *SA* sends messages to addresses received from the *LOA* and requests information about local requirements
3. Foreign *LOA*s (*HOA*s) reply informing about their policies
4. *SA* performs multicriterial optimization and selects one or more of the available universities as the place where the student will go for the exchange
5. *SA* informs student about possibilities and suggests which one to choose
6. *SA* informs its home *LOA* about the place selected for the student to go to
7. *LOA* informs the *SA* if student qualifies for the exchange – if student did not qualify, the *SA* goes back to 5. and the process repeats
8. Home *LOA* sends all of the necessary documents for the student to become a part of the exchange program to the host *LOA* (*HOA*) and obtains confirmation
9. *LOA* informs *NA* that a given student will be participating in a given student exchange
10. *HOA* registers an incoming exchange student (her/his *SA* is also registered with the local system)
11. *SA* moves to the foreign host
12. *SA* contacts *HOA* about address of appropriate *DA* for the field of study
13. *SA* contacts appropriate *DA*
14. *DA* informs the *SA* about courses available

15. *SA* performs multicriterial optimization and on the basis of knowledge of student preferences (e.g. student is a night owl) selects courses that match these preferences (e.g. no courses scheduled to meet before 1PM)
16. *SA* informs the *DA* that student completed scheduled courses
17. *DA* informs the *SA* and the *HOA* how many ECTS student accumulated
18. *HOA* "allows" the *SA* to go home
19. *SA* moves to its home container
20. *HOA* informs *LOA* about results of student exchange program participation (grades, ETCS, etc.)

Obviously, at this stage of the project the multicriteria decision making processes, mentioned above in points 4 and 14, has been replaced with a set of very simplistic selection procedures. However, delving into decision making was not of our current interest and is definitely outside of the scope of this paper. What we were interested was to develop the system skeleton and illustrate experimentally that it works. To show that agents communicate accordingly to the specification and that agent mobility is appropriately utilized to work in unison with proposed student mobility. As illustrated in the next section, we have fully achieved this goal.

3. System implementation and operation

The proposed system has been implemented in JADE 3.3 [2]. In a JADE based agent system, all agents exist within a platform that can be spread among multiple computers. Within a platform, agents reside in and move between containers. In our experimental setup, every container represents one university. We have inserted *SA*s, *LOA*s and *DA*s into each container (note that an *LOA* plays a role of a *HOA* depending on the direction of proposed student exchange). Additionally as first *NA* is created into the Main container.

The result of the system run was the *SA* performing all necessary steps to organize the exchange program for its student-master. In our current implementation we use a very simple selection criteria and the place where the exchange program was to take place was selected on the basis of only two student preferences: field of study and number of ECTS credits she gathered thus far. An example of a system run is presented in Figure 4 (here the, JADE provided, *Sniffer Agent* which "records" all messages incoming to and originating from agents, it was told to "sniff," was used to depict the operation of the system). In this example we observe a sample scenario involving three universities (located at three separate computers): Technical University of Gdańsk, University of Southampton and University of Edinburgh. At the Technical University of Gdańsk *DA*s representing Mathematics and IT departments have been created. Similarly, at the University of Edinburgh we see departments of Philosophy and Biology, while at the University of Southampton, departments of Chemistry and IT. Furthermore, a *SA*s were also created within containers representing the Technical University of Gdańsk and the University of Southampton (however in this example, the *SA* at Southampton is passive). Finally, the Main container hosts the *NA* (named *df*) and the *GUI Agent*. The main point of this scenario is for an IT student at the Technical University of

Gdańsk to arrange (and complete) an exchange with the IT department at the University of Southampton and this mission is accomplished.

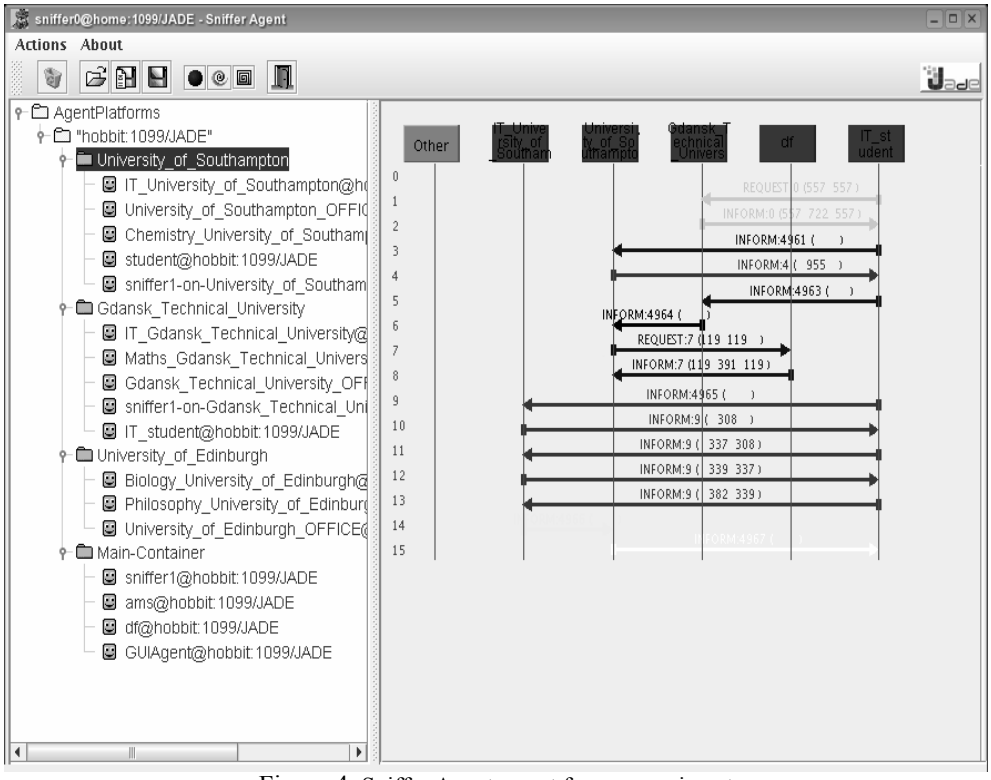

Figure 4. Sniffer Agent report for an experiment

In a separate experiment, using the ***psexec*** scripting program [4] we have created 22 containers representing 22 universities located in 22 countries, on 20 separate, networked computers. We have then placed "random" departments on each one of them and successfully run experiments with "students" (*SA*s) seeking exchange programs among all of these computers (university departments). More details of these experiments (involving an earlier, somewhat less sophisticated version of the system) can be found in [1].

4. Concluding remarks

Our project, in its current stage, illustrates the most important (from the point of view of agent system design and implementation) features of system that would enable student mobility automation. Those are: mobility, communication, registration, searching etc. Furthermore, the system skeleton has been implemented and shown experimentally to work (even though, we have to admit, utilizing an extremely simplified sets of rules for decision making, selection etc.). We were also able to scale the code to 20 computers hosting 22 containers representing 22 universities with multiple departments. This was achieved on a network of PC's

with AMD Duron processors running at 1,2 GHz, each with 256 Mbytes of memory connected via w 100 Mbit/s switch. This indicates that the proposed system can be expected to be quite scalable overall.

One of the important things to consider when constructing agent systems is that each such a system has to reflect the real world. Our example shows potential of software agents to automate an existing real-world scenario. In the next steps of the development of this system, we will attempt at making it to resemble the reality even more, by focusing on developing and implementing the following features:

1. Student Agent personalization (agent that actually know what the student really "wants" and is able to truly represent her interests)
2. Adding functions to the Department Agent, which extends the communication between *DA* and *SA*.
3. Adding a more intelligent decision making components, where the selection of a given university, or courses to take at the host university are based on a realistically selected set of criteria
4. Making communication between agents more realistic by (developing and/or utilizing existing ontologies and negotiation protocols).

Moreover, performing international tests (computers located in different countries) is compulsory as what we want to achieve is globally working system. We will be reporting on our progress in the near future.

References:

1. Ganzha M., Kuranowski W., Paprzycki M., Rahimi S., Szymczak M., Designing an Agent-based Student Mobility Support System. In: *Proceedings of the Information and Communication Technologies International Symposium*, Tetuan, Marocco, to appear
2. JADE. Java Agent Development Framework. See: http://jade.cselt.it
3. Preist C., Byde A., Bartolini C., Piccinelli G., Towards Agent-Based Service Composition Through Negotiation in Multiple Auctions. Trusted E-Services Laboratory, HP Laboratories Bristol, HPL-2001-71 (R.1), 23.05.2002.
4. PS Tools. http://www.sysinternals.com

Maria Ganzha
Prywatna Wyższa Szkoła Zawodowa w Giżycku,
ul. Daszyńskiego 9, 11-500 Giżycko, Poland, Tel.:(0-87) 428-89-30
e-mail: ganzha@pwsz.net

Wojciech KURANOWSKI
Prywatna Wyższa Szkoła Zawodowa w Giżycku,
ul. Daszyńskiego 9, 11-500 Giżycko, Poland, Tel.:(0-87) 428-89-30
e-mail: wojtek@pwsz.net

Marcin Paprzycki
Computer Science Department, Oklahoma State University, Tulsa, Oklahoma, USA,
Computer Science, SWPS, ul. Chodakowska 18/31, 03-815 Warszawa, Poland
e-mail: Marcin.Paprzycki@swps.edu.pl