Designing an Agent-based Student Mobility Support System

Maria Ganzha [1], Wojciech Kuranowski [1], Marcin Paprzycki [2,3], Shahram Rahimi [4], Michal Szymczak [5]

[1] *Department of Computer Science, Gizycko Private Higher Educational Institute, Daszynskiego 9, 11-500 Giżycko, Poland*
[2] *Computer Science Department, Oklahoma State University, Tulsa, Oklahoma, USA*
[3] *Computer Science, SWPS, ul. Chodakowska, Warszawa, POLAND*
[4] *Department of Computer Science, Southern Illinois University Carbondale, Illinois, USA*
[5] *Department of Mathematics and Computer Science, Adam Mickiewicz University, ul. Umultowska, Poznań, Poland*

Emails: ganzha@pwsz.net; wojtek@pwsz.net; Marcin.Paprzycki@swps.edu.pl; rahimi@cs.siu.edu; wildmike@interia.pl.

ABSTRACT - There exist a number of areas where autonomous software agents are expected to play an important role as a new generation of tools facilitating efficient management of information. One such area that was not considered thus far is support for student mobility. While not much of an issue within the Unite States, student mobility is one of the important objectives within the European Union. In this paper we present an initial design of an agent system that supports this purpose.

1. 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 mobility of students that is supported financially by the EU through programs like Socrates (and, in the near future, Mundus). Such programs allow students studying in participating institutions of higher learning, based on bilateral (university-to-university) agreements, and on certain qualifying procedures, to visit universities in other EU countries and spend there one or two semesters, while obtaining a living stipend from the EU. Obviously, to be able to arrange such a visit a number of administrative steps need to be completed (further steps are also required post completion of an exchange). Completion 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 discussed in this note involves development of an agent system that would provide support for entities involved in student mobility programs (both students and university administrators).

We proceed as follows, in the next section we summarize steps that area usually to be undertaken by a student who would like to participate in a mobility program (including the administrative infrastructure involved in reaching this goal). We follow with the description of the design of an agent system and a few details of its implementation (Section 3). In Section 4, we illustrate the performance of the system by describing two experiments; one involving three countries and one simulating twenty two countries. We complete this note with a brief description of our future research directions (Section 5).

2. STUDENT MOBILITY

Let us consider typical institutions of higher learning that are to be involved in an international student exchange program. Currently, when the flow of information supporting such a program is considered, we have teachers, students and university employees that have computer accounts somewhere within the university computer network, access to appropriate data repositories. These individuals communicate directly via e-mail and utilize more or less arcane ways of sharing necessary resources, e.g. applications, data stored in various databases etc. Some schools provide also an interface for students, which allows them to check current study-related issues, such as: schedule of selected courses, upcoming exams, gained credits etc.

Let us consider situation when a student from a given university located in an EU country wishes to participate in a student exchange program. In this case the following steps have to be completed:

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

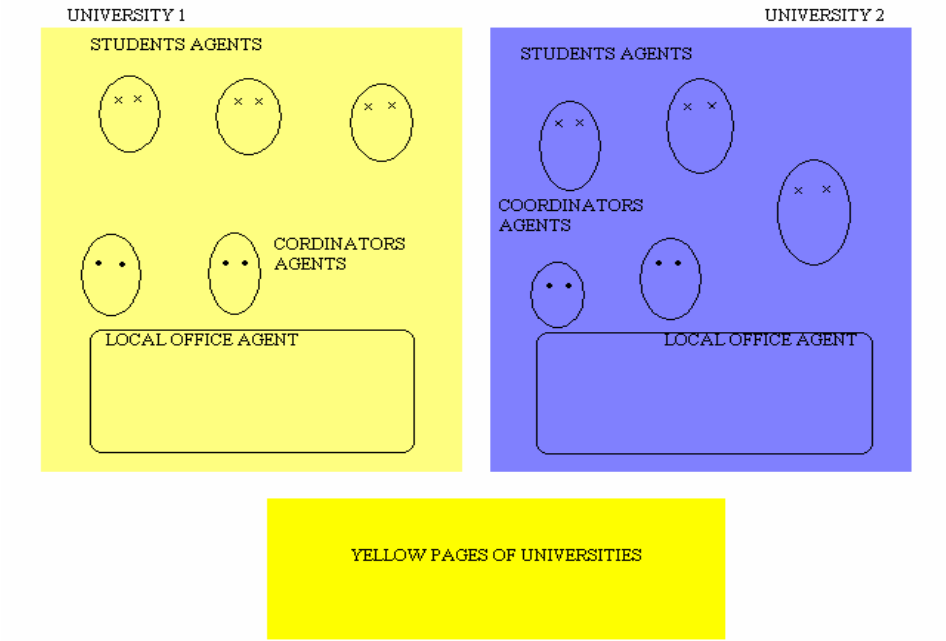

Figure 1. General System Architecture

after arrival at foreign university:

6. Contacting local coordinator
7. Arranging the schedule of courses
8. Controlling information about 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.

The first four steps involve mostly cooperation between the student and Dean's Office at her local university, as well as an information exchange with local program coordinator. Step five is often completed "automatically" by established office at the host institution that receives information about incoming exchange students as a part of the document circulation involved in steps 1-4. In the remaining cases 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 and utilize his/her knowledge 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 leaving and incoming exchange students, steps 1-4, 6-8 and 9 don't present any problem **when considered independently**. Problems start to materialize when all these steps have to be completed "together" and thus, when data (appropriate documents of various types) has to start circulating between different units within a given university; between different units in different universities and, finally, between these units (both local and foreign) and the student. Let us stress that in each case a bi-directional transfer of documents is involved. 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) or numerous telephone calls.

Our system has been developed in response to these challenges and is based on experiences gathered by of one of the authors (MS) who has recently spend a year as a Socrates funded exchange student in Finland.

3. AGENT SYSTEM – TOP LEVEL DESIGN

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" by the interested parties. We propose a system for automatic cooperation of students with their university administration workers and also between the different schools offices that will facilitate the flow of information required to set an exchange program in motion. The top level architecture of the system has been depicted in Figure 1. Here, we divided the functionality into two parts (according to existing communication patterns):

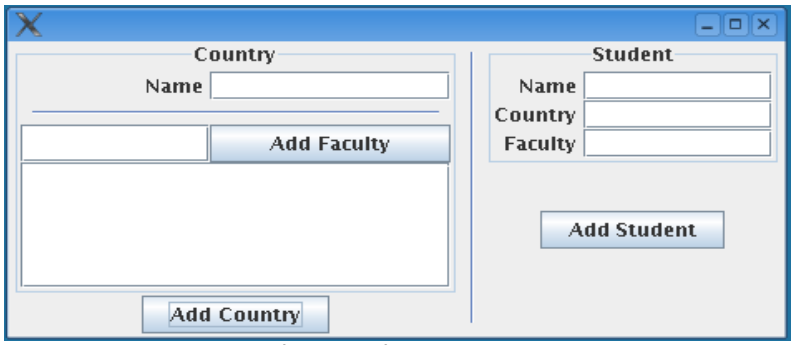

Figure 2. The GUI Agent

1. Local, intra-school communication

2. Global, inter-school communication.

The first part assumes that every student has its personal Student Agent (SA) with which she communicates via a PC connected to the network (or any mobile device (M. Gordon, M. Paprzycki, M. Gawiniecki, P. Kaczmarek, to appear)). She can organize or view her schedule, check the total number of credits acquired thus far, make an appointment with her professor and/or advisor etc. The SA, among others, should minimize the number of student visits that the student has to pay to various offices, and in this way to make life easier for both herself and for the administrators.

The next important feature of the system are Coordinator Agents (CA) that may be conceptualized as virtual-heads of individual departments of a particular university. These agents are envisioned to be responsible for courses, schedules and research programs of departments which they represent.

The Local Office Agent (LOA) is a simple agent that stores information about students and departments. It also holds addresses of SA's and of Coordinators Agents.

This last agent, while quite simplistic, plays a crucial role in the second part of our system. Here the contact data of LOA is stored in Yellow Pages service point, which should be understood as a database of agent addresses (C. Preist, A. Byde, C. Bartolini, G. Piccinelli, 2002). As we want to support student exchange programs, the SA's have to be able to find data about foreign schools that are available to students interested in participating in such a program. (Note that the level of usability of delivered information would increase with introduction of content personalization to each SA, however, developing such capabilities is beyond the scope of this note.) When the university where the student is going to visit is selected, the SA provides details about studying in a particular institution and helps organizing necessary documents. After all documents are collected, the SA delivers them to the Local Office Agent, which, in turn forwards them to the LOA of the Host institution. After student arrives at the Host university, the SA continues working. It communicates with the Host office and the local program coordinator and supplies the exchange student with all required information.

3.1. Why Agents?!

Issues listed in the first paragraph of Section 2, require support of bidirectional exchange of data between different universities as well as between institutions and a student, both locally and globally. Additionally we need the student-system interface to be mobile.

Agent based systems are designed to deal with those situations. First, well designed inter-agent conversations, negotiation protocols, languages and ontologies allow efficient communication between sides without sending faxes and letters or making appointments. In case of distinct data formats, when exchanging data between Local Office Agents, they can be equipped with proper parsers.

When spatial mobility is concerned, agent technology becomes a natural solution, as agent migration is one of the main features of Multi Agent System. Let us suppose that owner of a particular SA is flying from Lisbon to Paris for one year of studies. In that case his SA migrates from the Lisbon University Agent Container into the University of Paris Agent Container. After coming to his new school our student may login again, while his SA is already hosted locally.

4. IMPLEMENTATION

The proposed system has been implemented in JADE 3.3 (JADE). We have constructed four types of agents:

1. Office Agent – OA,

2. Coordinator Agent – CA,

3. Student Agent – SA,

4. GUI Agent – GA.

They are the main "characters" of the scenario described above. To illustrate how our system works, let us proceed on the basis of a simplified

scenario, and illustrate it by screen-shots of the running system.

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. Let us report on two experiments we have run. In the first one, we have used the GUI agent (Figure 2) to create three containers located on three separate computers (emulating a realistic scenario, where separate universities have separate computer systems).

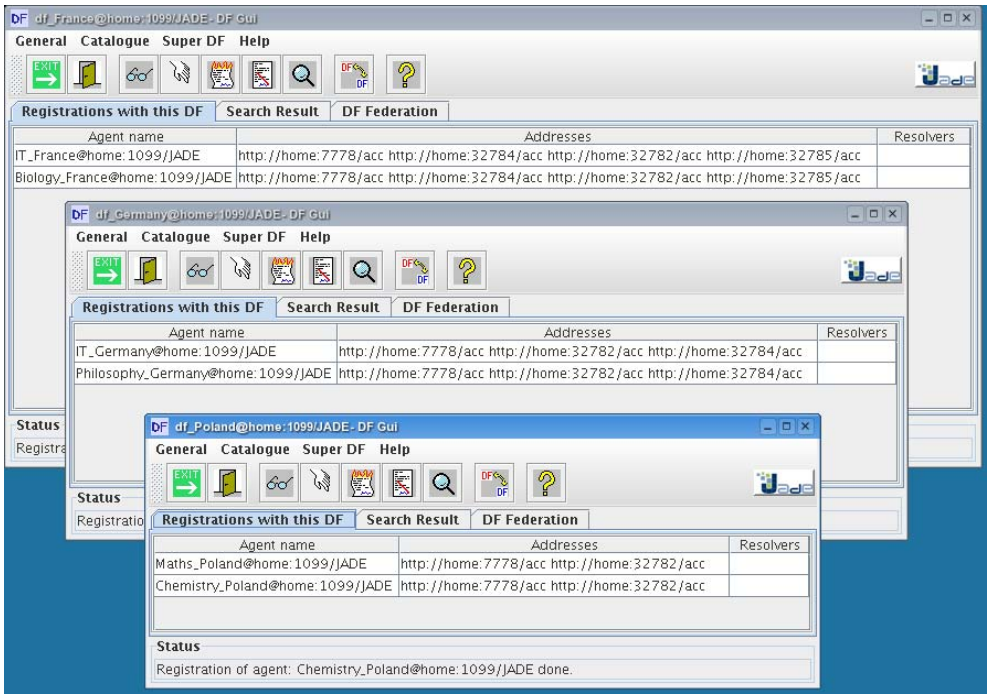

Figure 3. Containers representing Polish, German and French Universities.

We have then inserted OAs and CAs into each container. In this experiment we have created two CAs within each container. They represented Mathematics and Chemistry departments in the case of a Polish university, IT and Philosophy departments in a German university, and Biology and IT departments in a French university. Finally the SA, which plays the role of a Polish student, was started in the container representing the Polish university. The system as created is depicted in Figure 3.

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 illustrate the operation of the system).

To further illustrate capabilities of our system we have created twenty two separate containers which represented universities located in 22 EU courtiers. These containers could have been created individually via the GUI agent (Figure 2). However, to simplify our work we have created them utilizing a script implemented using the psexec program (PS Tools). Thus, for creation of the Poland-container we have used command:

```
psexec \\k03s12 -i -d -c -f -w c:\jade c:\jade\x.bat Poland F1 F2
```

where: \\k03s12 is the name of the computer residing within the local network (and this name varies), and x.bat is a standard batch-file for starting Agents in the JADE-environment:

```
runjade.bat -host 10.0.12.1 -container -container-name %1 df_%1:examples.yellowPages.SubDF %1_OFFICE:Socrates.Office("%2","%3")
```

As previously, each of these 22 containers was created on a separate computer.

Every agent, immediately after being created, registered itself within the Yellow Pages Agent, also called the directory facilitator (df). Each container has its own df, here df_Sweden, df_Slovakia and so on. Additionally there is one Main Yellow Pages Agent created every time the JADE platform is started. CAs and the SA register at their local df, whereas OAs perform registration at the Main df. Figure 3, while depicting the system created in the first experiment, represents

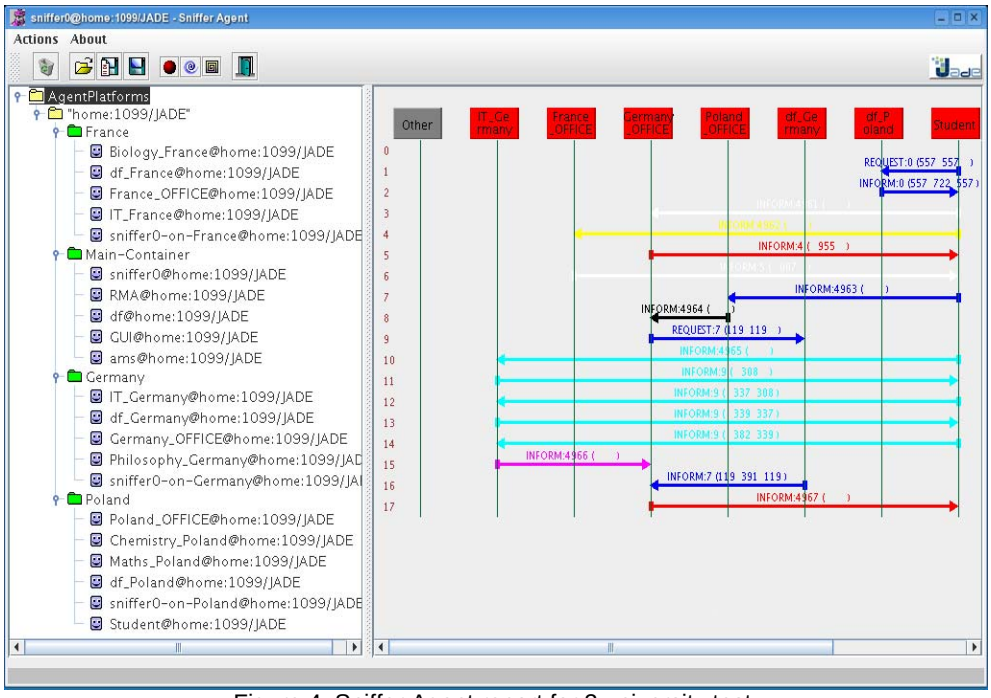

Figure 4. Sniffer Agent report for 3 university test.

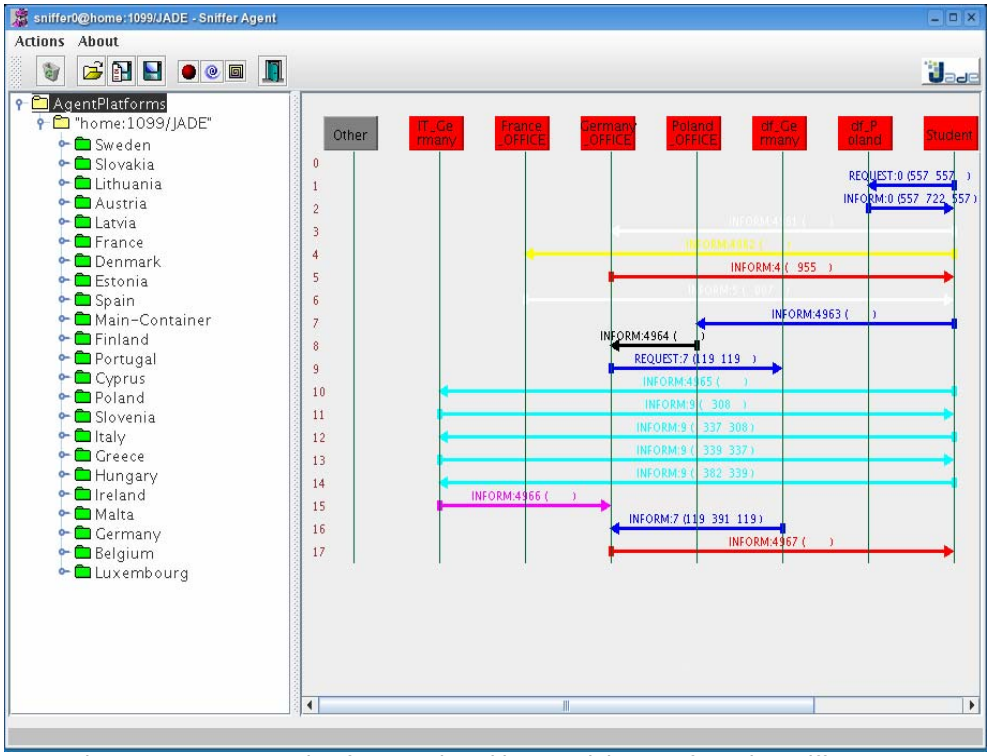

Figure 5. System running for 22 universities; partial report form the sniffer agent.

also information stored in the central df (in the case of 22 universities, each of them would be represented similarly). Figure 5 presents the system set up in the second test, with the 22 “countries” visible.

Let us now present a complete list of interactions between various agents that take place when the SA attempts at arranging the exchange program for the student. We assume here that the system has been initialized and that the student has communicated with its SA and established the program selection criteria (e.g. country, subject area, number of ECTS credits etc.). Then, the system starts to operate and performs the following actions (it is assumed that the system performs all of the actions completely autonomously and that the student specifies only initial requirements and then agents make all of

the decisions):

1. SA sends search request to the df to get addresses of all university department coordinators in the field of study specified by the student in the original request

2. The SA sends ACL messages to all addresses received from the df (local CAs) and requests information about local requirements

3. CAs reply according to their policies

4. SA performs a multicriterial optimization and selects one of the available universities as the place where the student will go for the exchange

5. SA informs its home OA about the place that was selected for the student to go to

6. Home OA sends all of the necessary documents for the student to become a part of the exchange program to the host OA

7. Host OA registers this SA as an incoming exchange student at local df

8. SA moves to the foreign host (see Figure 5)

9. SA contacts foreign host CA

10. CA informs the SA about courses available

11. 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 meeting before 1PM).

12. CA informs the SA and host OA how many ECTS SA is going to accumulate

13. SA informs the CA when the student completes the courses

14. CA informs about it the host OA

15. OA "allows" the SA to go home

16. SA moves to its home container

Obviously, at this stage the multicriteria decision making processes have been replaced with a very simplistic set of selection procedures. However, delving in issues involving decision making was not of our current interest and is outside of the scope of this paper. What we were interested was to develop the system skeleton and illustrate experimentally that it works, 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 Figures 3, 4 and 5 this goal has been achieved.

5. CONCLUDING REMARKS

Our project in its current stage illustrates most important (from the point of view of agent system design and implementation) features of system that would enable student exchange program automation. Those are: mobility, communication, registration, searching etc. Furthermore, most important parts of the system have been actually implemented and shown experimentally to work (even though, we have to admit, utilizing an extremely simplified sets of rules for decision making, selection etc.).

The most important thing to remember when constructing agent systems is that each such a system has to reflect the real world. Our example shows ability 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 its student-master "wants" and is able to truly represent her interests)

2. Adding a more intelligent decision making component, where the selection of a given university is based on a realistically selected set of criteria

3. Communication between Office Agents and already existing data repositories at universities (in the initial stage add the database component to each "university" and instantiate utilization of documents stored there)

4. Making communication between agents more realistic by (developing and/or utilizing existing ontologies and negotiation protocols).

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

References:

C. Preist, A. Byde, C. Bartolini, G. Piccinelli, TowardsAgent-Based Service Composition Through Negotiation in Multiple Auctions. Trusted E-Services Laboratory, HP Laboratories Bristol, HPL-2001-71 (R.1), 23.05.2002.

M. Gordon, M. Paprzycki, M. Gawinecki, P. Kaczmarek, The Problem of Agent-Client Communication on the Internet, PDCP, to appear.

JADE. Java Agent Development Framework. See: http://jade.cselt.it

PS Tools. http://www.sysinternals.com