Designing and Agent-based Student Mobility Support System

MARIA GANZHA, WOJCIECH KURANOWSKI
Department of informatics
Gizycko Private Higher Educational Institute
Daszynskiego 9, 11-500 Gizycko
POLAND

MARCIN PAPRZYCKI
Computer Science Department
Oklahoma State University
Tulsa, Oklahoma, USA
and
Computer Science
SWPS
ul. Chodakowska, Warszawa, POLAND

SHAHRAM RAHIMI
Department of Computer Science
Southern Illinois University
Carbondale, Illinois, USA

MICHAL SZYMCZAK
Department of Mathematics and Computer Science
Adam Mickiewicz University
ul. Umultowska, Poznan, POLAND

Abstract: - There exist a number of areas where autonomous software agents are expected to start playing an important role as new mechanisms facilitating efficient management of information. One such area that was not considered in the past is support for student mobility. While not so much of an issue within US, student mobility is one of the important goals within the European Union. In this paper we present an initial design of an agent system that supports this goal.

Key-Words: - Agent systems, student mobility, mobile agents, negotiations, JADE

1 Introduction

One of the important goals that the European Union is striving ad achieving is social mobility. One of the interesting aspects of this goal is mobility of students, supported financially by the EU through programs like Socrates (and in the near future, Mundus). The idea is the following. Based on bilateral (university-to-university) agreements, and on certain qualification procedures, students can visit universities in other EU countries and spend there one or two semesters, while obtaining a stipend from the EU. Obviously, to be able to arrange such a visit a number of administrative steps need to be completed. Our work involves development of an agent system that would provide support for students wanting to participate in student mobility programs.

In the next section we summarize the steps necessary to be undertaken by a student to participate in a mobility program. We bring these steps into the context of administrative infrastructure involved in reaching this goal. We follow with the general design of the system (Section 3). And complete this paper with information about the current status of the implementation of the system (Section 4) and future research directions (Section 5).

2 Student Mobility

Level of automation of information flow concerning formalities involving most institutions of higher learning, which exchange their students, is as follows. Teachers, students and university administration workers usually have their own accounts on university network drives, and communicate via email and share resources, e.g. applications, data bases. Some provide also interface

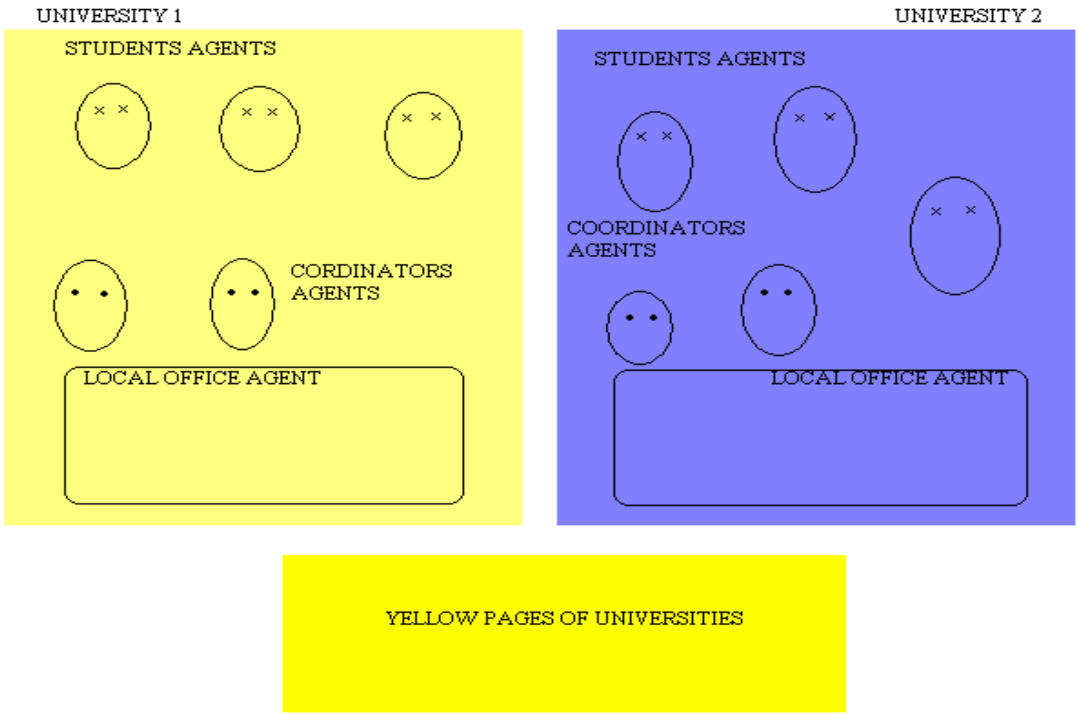

Fig. 1. General System Architecture

for students, which allows them to check current studying issues, such as: schedule of selected courses, nearest exams, gained credits etc.

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

before departure:

1. Choosing foreign university
2. Realizing its requirements
3. Realizing all requirements of her/his own university
4. Delivering all necessary data to both its own and the foreign university
5. Organizing a place in dormitory

after arrival:

6. Contacting with local coordinator
7. Arranging the schedule of courses
8. Controlling information about courses and credits and credits to meet exchange program agreements

after returning to the home teaching unit:

9. Completing a survey or delivering a report to the home coordinator.

The first four steps involve mostly cooperation of student with her/his home Dean's Office and information exchange with the program coordinator. Step five is done automatically by some of the host administration workers who receive information about coming students from their home units. In other cases student has to search a flat on his own or communicate with separate organization which supervises dormitories. After moving to the chosen school our individual stays in contact with the host coordinator.

In the universities appropriately prepared to handle incoming exchange students, steps 1-4, 6-8 and 9 don't present any problem when considered independently. The problem materializes when the data between different teaching units and student has to be transferred. Moreover not even every university has the system of student electronic data transfer support, hence another set of inconveniences. As a result an extremely large number of documents has to be transferred, which involves sending letters, faxes, receipts (in case of organizing a flat) or telephone calls. Our system has been developed in response to these challenges and is based on experiences of one of the authors (MS) who spend a year as an exchange student in Finland.

2 Agent System – Top Level Design

The main idea of our project is to bring a solution which would make formalities of taking part in a Student Exchange Program simpler, also reducing number of cases which presently have to be dealt directly by the interested sides. We propose a system for automatic cooperation of students with their university administration workers and also between the different schools offices. The top level architecture of the system has been depicted in Figure 1.

Here, we divided the functionality into two main parts:

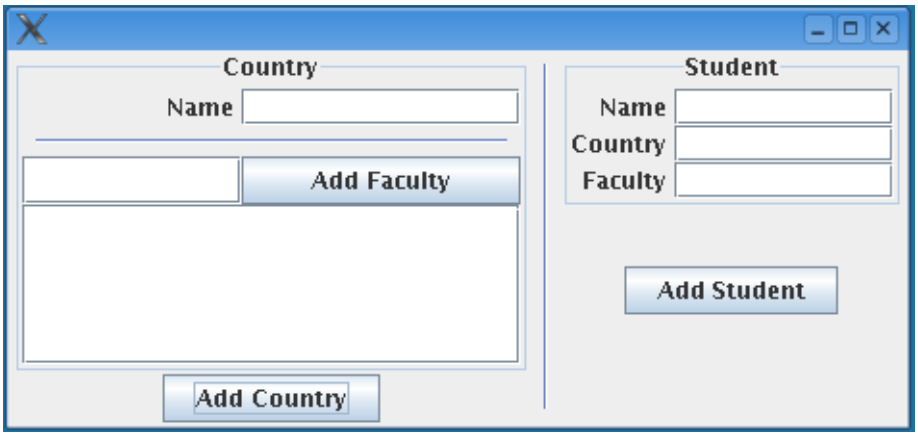

Figure 2. GUI Agent

1.Local, intra-school communication.
2.Global, inter-school communication.

First part assumes that every student has its Personal Student Agent (PSA) with which he communicates via PC connected to the network or a mobile phone. She can organize or view her schedule, check amount of credits secured, make an appointment with her teacher etc. It should minimize the individual student visits paid in the offices, i.e. make ones' life easier.

Next important feature are Coordinator Agents (CA) that may be understood as heads of all the departments of particular university. They are responsible for courses, schedules and research programs of the department which they represent.

Local Office Agent (LOA) stores information about students and departments. It also holds the addresses of PSA's and Coordinators Agents.

This last feature plays the main role in the second part of our system ([1]). Here contact data of LOA is being stored in the Yellow Pages service point, it easiest to understand is a database of addresses. As we want to support student exchange programs the PSA's are able to find data about interesting foreign schools. The level of usability of delivered information would increase in case personalizing each PSA. Later PSA informs his owner about details of studying in particular institution and helps organizing documents. After making decision by the interested individual the PSA inform students Local Office Agent, this one sends student data to the Host LOA. After moving to the host institution PSA of our student still works, it communicates with the Host office and coordinator and supplies the exchange student with all information required.

2.1 Why Agents?!

The problems listed in paragraph 1 of this section demands support of exchange of data between different universities as well as between institution and a student, both in local and global ranges. Additionally we need our student interface to be mobile.

Agent based system deals with those. Firstly, well designed inter-agent conversation, negotiation protocols, languages and ontologies allow efficient communication of interested sides without sending any faxes and letters or making any appointments. In case of distinct data formats when exchanging data between Local Office Agents they can be equipped with proper parsers.

When it comes to traveling of our student agent technology brings us real benefit, as agent migration is one of the main features of Multi Agent System. Lets suppose that owner of particular PSA is flying from Lisbon to Paris for one year of studies. In that case PSA migrates from Lisbon University Agent Container into University of Paris Agent Container. After coming to his new school our student may login again, but this time on the host institution.

3 Implementation

The proposed system has been implemented in JADE 3.3. ([2]) 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 the following scenario and illustrate it by the screen-shots of the running system.

In the JADE 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. To illustrate capabilities of our system we have created three separate containers which represented Polish, German and French universities. These containers were created via the GUI provided by the specially designed *GUI* agent (Figure 2). Each of these containers was created on a separate computer (emulating a realistic scenario, where separate universities have separate computer systems). Within each of these containers we have inserted OAs and CAs.

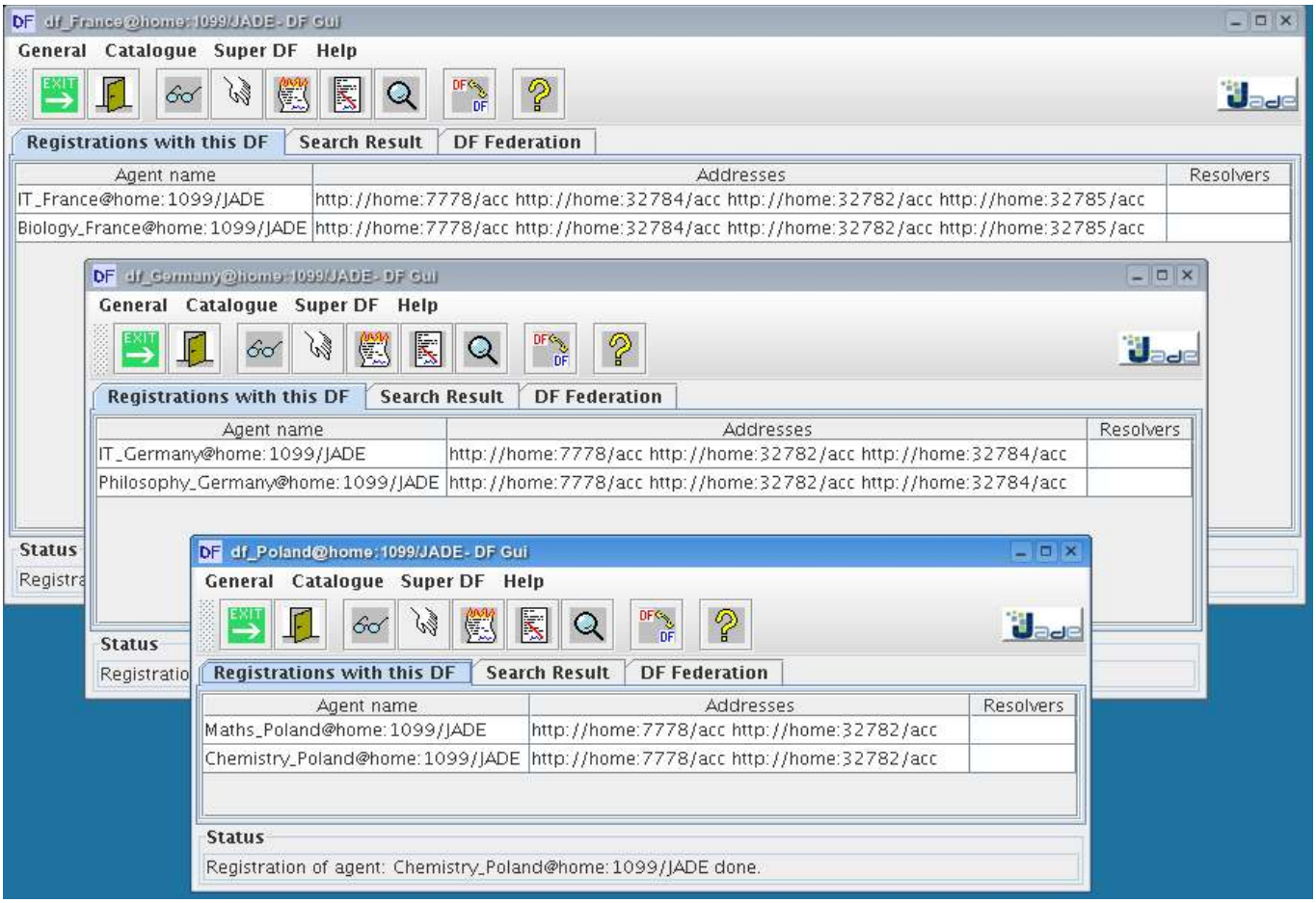

Figure 3. Yellow Pages Agent

For the sake of this test there were two CAs on each container. They represented Maths and Chemistry departments in case of the 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 SA has currently two main characteristics, which are: his field of study and number of ECTS credits he gathered thus far.

Every agent, immediately after being created, registers itself within the Yellow Pages Agent, also called the directory facilitator (df). Each container has its own df, here df_Poland, df_Germany and df_France. Additionally there is one main Yellow Pages Agent created with every start of the JADE platform. CAs and SA register at their local df, whereas OAs perform registration in the main df. Views of registered agents at local dfs tables are shown in Figure 3

JADE platform is equipped in number of standard agents. One of the used by us is the sniffer agent which "records" all messages incoming to and coming out from agents it is told to sniff to. We will explain details of our implementation and the way it simulates student exchange program procedures here using sniffer agent window (v. Figure 4):

1.SA sends search request to df to get addresses of all university department coordinators of his field of study.
2.He sends to all the received addresses request of requirements.
3.CAs reply according to their policies.
4.SA chooses one of the universities possible.
5.SA informs his home OA about the place he wants to go to.
6.Home OA sends data about the student to the host OA.
7.Host OA registers this SA as exchange student at local df.
8.SA moves (v.Figure 5).
9.SA contacts his host CA.
10.CA informs about courses available.
11.SA chooses courses.
12.CA informs SA and host OA how many ECTS SA is going to get.
13.SA informs CA when he completes courses.
14.CA informs about it host OA.
15.OA allows SA to go home.
16.SA moves to his home container.

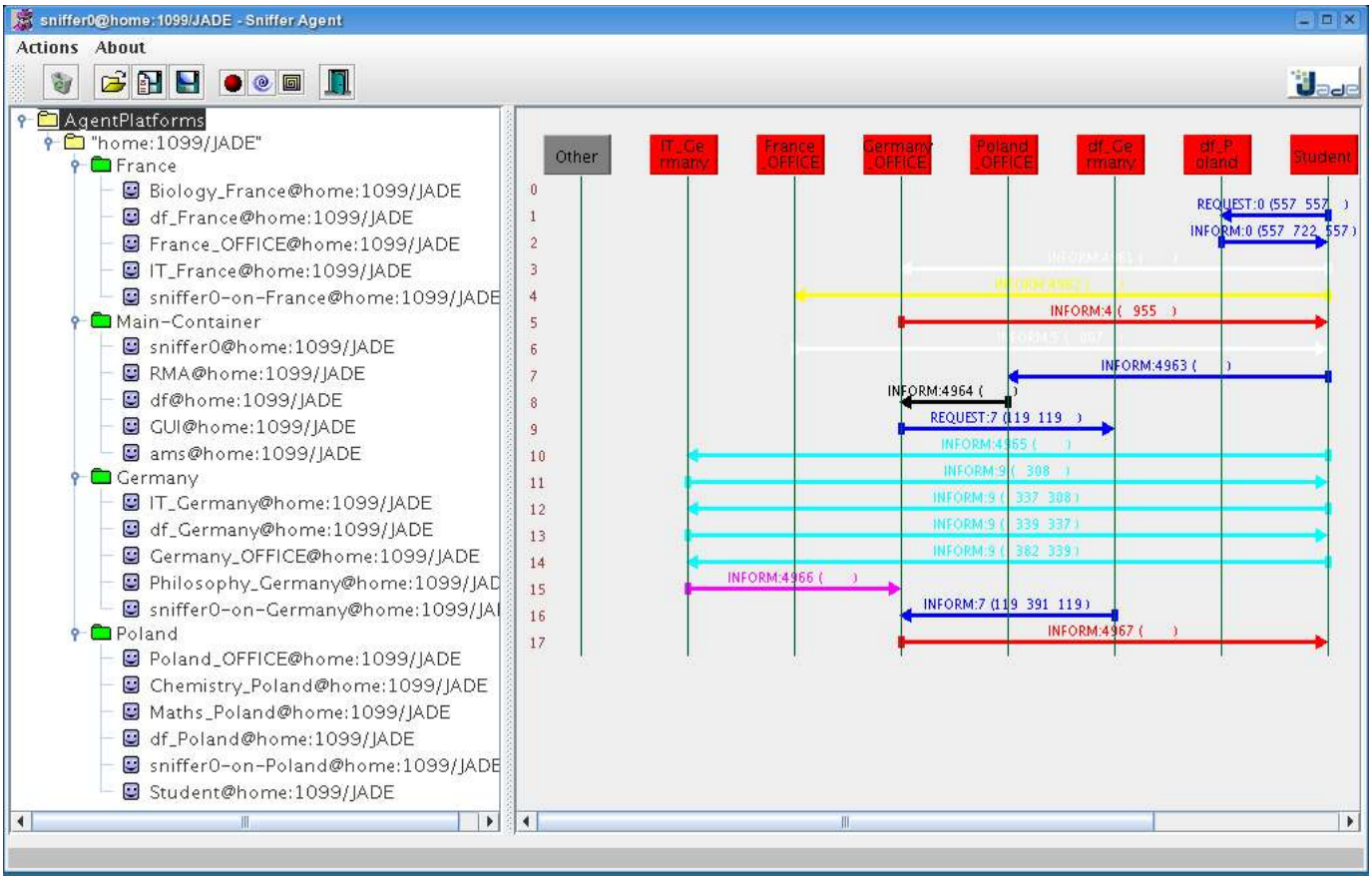

Figure 4. Sniffer Agent Window

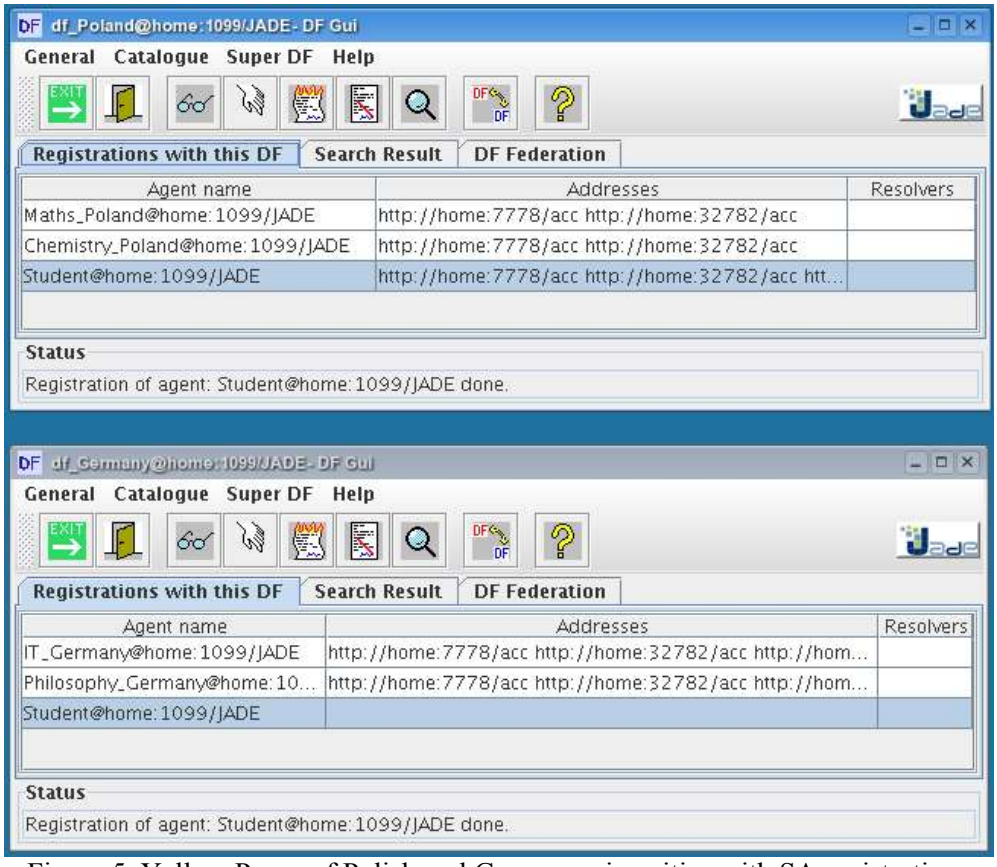

Figure 5. Yellow Pages of Polish and German universities with SA registrations

4 Conclusion

Our project at it current stage shows certain features of system which would enable student exchange program automation. Those are: mobility, communication, registration, searching and most of all combining listed characteristics.

The most important thing to remember when constructing multi agent system is that it has to reflect real world system. Our example shows ability of automation of the real world scenario. If we want it to resemble more reality following features should be implemented:

1. Student Agent personalization,
2. Communication between Office Agents and already existing data repositories at universities,
3. Communication improvement between agents (developing ontologies and negotiation protocols).

Moreover, performing international tests is compulsory as what we want to achieve is globally working system.

References:

[1] 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.

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