Improving Study Planning with an Agent-based System

Aki Vainio and Kimmo Salmenjoki
Department of Computer Sciences
Faculty of Technology, University of Vaasa
Box 700, Vaasa, Finland
E-mail: aki.vainio@uwasa.fi, ksa@uwasa.fi

Keywords: autonomous agents, personalized study plans, collaborative planning

Received: May 8, 2005

This paper presents a system for designing and updating a personalized study plan in a collaborative environment. Unlike existing systems, which are mainly interested in storing the study plan, this system based on learning agents is able to suggest a study plan and if needed, identify potentially problematic choices in the future, thus bringing dynamics in to the system. By collaborating with other agents in a multi-agent environment, the chances of finding a mutually beneficial result is improved. A prototype of the system for creating study plans is available. Initial empirical results show that after a short learning period, the system is able to form a study plan which requires minimal attention from the students.

Povzetek: Predlagan je sistem učenja s pomočjo agentnega sistema.

1 Introduction

The Bologna process imposes many changes to the system of higher education in Europe [12]. In Finland, one of the more recent changes is the new limit to the amount of years a student can study. Currently, it is common to study for six or more years to obtain a master's degree. In the future, five years should be the norm. Since the students, university personnel or the ministry of education are willing to relinquish any of the requirements for a master level degree, better planning is needed.

According to [1], one tool for achieving this planning is the personalized study plan, which is a requirement for all students in Finland in the future. Since a first year student does not have enough knowledge to make decisions for the whole span of his studies, creating and keeping the personalized study plan up to date is important as the student becomes more knowledgeable. [1]

Current systems are not using the full potential of advanced information systems. The approaches OVI [2] and Oodi [3] are attempts at simplifying the process, but neither seems able to address the personal study planning as all they do is simply store the study plan. OVI even requires the student to make all the course choices, even though the mandatory courses could easily be selected beforehand. Oodi will include such functions and according to the current design, it can also check the study plan for correctness, but the timetables for the Oodi project are such that a usable version will not be available for at least a few years.

The new system proposed here will be a part of Wompat-system [4], which is designed as a tool for students to use when planning their schedules. Wompat has been used in University of Vaasa for two years with good feedback from both administration and students. Also, the consortium behind the Oodi system has expressed interest in it.

The student view of the web based Wompat system is shown in figure 1 with course options on the left, schedule organized by weeks, days and hours in the center and courses chosen by the student in the right. Using Wompat, the student obtains a weekly schedule for all the courses that he or she intends to study.

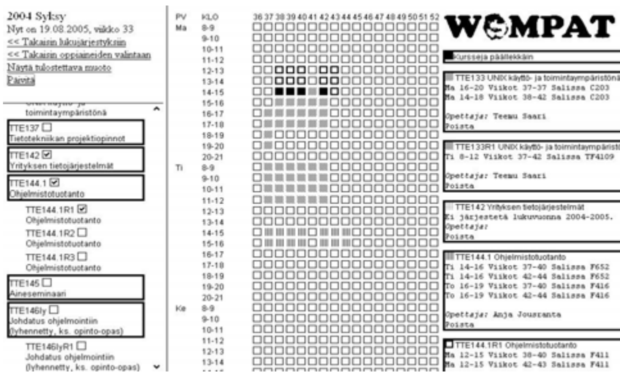

Figure 1. Screenshot of the current Wompat-system

2 Automated Approaches for Study Plan Formulation

2.1 Agents

According to [7] a rational agent is described by its PAGE-description. The PAGE-description consists of percepts, actions, goal and environment. The creation of the study plan can be clearly divided into two parts: selection of courses and timing of the studies taken by the student. According to this, our system will consist of two agents, where the first one controls the selection of courses, storing the selection in a database and the second agent uses these selections in order to automate the preferred annual schedule of the selected courses. Communication between the agents is handled by changing the environment, which in this case means changes in the database. The core software of the personalized Wompat system is a combination of the two agents and their automated communication via the information content of the database. The details of the database will be given in section 3.

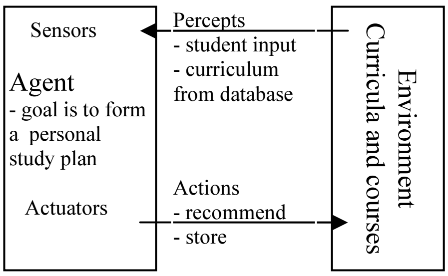

Figure 2: PAGE-description of the agents in Wompat

With this agent approach the use of study plans can be enhanced. One specific problem in small departments is the lack of resources for arranging courses annually. Often many courses, which are not popular, are only arranged every other year or even less frequently. In order to optimize the use of resources, the departments can use the information gained from the personalized curricula to arrange only the courses with enough interest.

The proposed system could also use this same information to examine beforehand which courses are likely to be arranged in a given year. If the system finds that a student has chosen a course with low common interest, it could make the student aware of the problem. The system could also try to find another course which might be of interest to the student, based on the student's earlier choices and the choices made in other curricula.

The system needs decision making when suggesting courses for a study plan and when finding the correct timing for a course. In many cases, the latter is not a real problem, but it is in some instances, where some courses might be suitable for two or more years.

Finding the courses, which are probably not arranged, is simple. The system can find them by simply checking how many people have chosen a course on a given year. The system can either be equipped with the knowledge of how many students are usually required for a course to be held. It can also be equipped with the knowledge of how many of the courses will be arranged in a given year and make an estimation based on that information.

The decision making in the agents of the system is based on knowing the prerequisite courses required for more advanced courses, the interests of the student and further learning of these issues. Through these methods, the system can find better suggestions to be presented to the student.

2.2 Prerequisite Sourse Utilization for Decision Making

The relationships of prerequisite courses and advanced courses form a directed acyclic graph (DAG). This graph can be used to find appropriate courses based on what the student has already taken or on the other hand, if the student or system sees fit to choose an advanced course, the system can easily find the prerequisite studies needed for that course and suggest them to the student. By using the DAG, the courses can be easily found by moving through the DAG recursively. On the other hand, if the student has not completed all the needed prerequisites and doesn't have enough time to complete them all, the system can suggest that the student stay away from those courses.

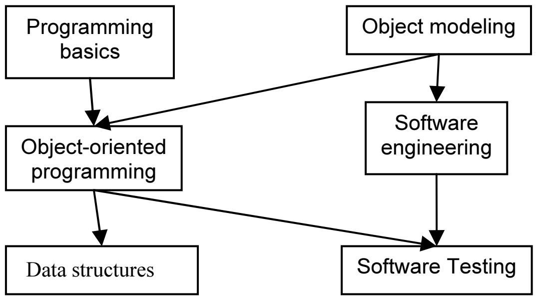

Figure 3: Example of sub-DAG of courses

2.3 Students Selection of Courses

As a student can focus his or her studies in a number of ways, certain prepared areas of orientation can be used to help the decision making. After these areas are defined, each course can be given a weighted relationship with these areas by an expert, for example a teacher.

To give the system information on which to base the decision making on, the student can give his or her interests in the orientation in a manner reminiscent of fuzziness, where the student can choose from descriptions acting as variables:

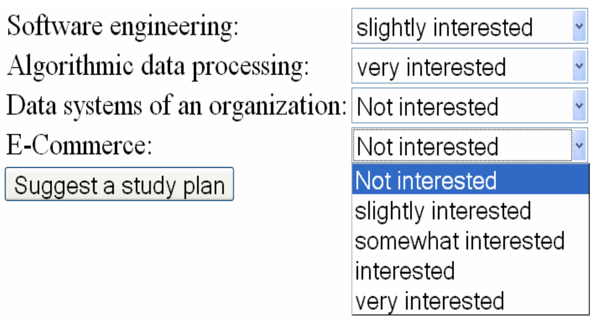

Figure 4: Form for gathering data on student's interests. See [5] for more details.

The agents receive the user's selections via the database. The database also contains the decisions made by the other students. All this information is used by the agents in the automated selection process.

2.4 Learning in the Agent System

Machine learning is change in the system which results in better performance [10]. Better performance in an agent means that the frequency of right or good decisions grows over time [10]. In this system, the agents can be said to learn if they choose the right courses more often and are able to time courses with better accuracy over time.

Since the system should be able to function as autonomously as possible, learning is required to give them some freedom from the subjective or uninformed views of the people who give the agent the preliminary information on the relationships between the orientations and the courses. Since the system has access to full feedback from all the decisions it has made, in the form of whether or not the students follow the agent's advice, learning can be very fast. This is called active and supervised learning [7].

Learning can be based on simple decision theoretical analysis using frequency as a basis for decision. However, with small group sizes, this isn't always possible or the data is not accurate enough. For this reason, a set of values based on the preset relationships can be used to give the system some basis to work on when there is not enough gathered data to draw conclusions from. This also stops the system from learning too much from the first few students. By using this method, the system does not work on probabilities, but rather on approximations of probabilities.

If needed, the system can be taught in a supervised manner. By giving the system some sample personalized curricula, it can use those as cases to learn from.

After the system has been in use for a long period of time, it might not be able to learn as quickly as before. The need to learn is still there as the curricula, tastes of the students and other matters change over time. This problem can be overcome by simply introducing limits on the number of students used in the learning phase.

Learning has another important function: Often the courses have no set year for completion. Even if the window is usually only years in these instances, this information might make a difference in the decision making process. This information can be learned from averages, with some tolerance for error. This learning also removes the need for telling the system when courses should be completed by the student. The system can make those decision based on the prerequisite courses as described in 2.2 and through learning.

2.5 The Process of Making the Study Plan Automatically

The process of making the study plan requires actions from both the student and the agent. The role of the student is mostly as a control measure to see that the study plan is to the students liking. The student has the power to change as many of the decisions made by the agents, but if the agents work correctly, not many changes should be needed.

Figure 5 presents the process and the messaging between the participants. The process is started by the student, who requires a study plan. Upon logging into the system, the student can be identified and the right curriculum can be chosen. Based on the curriculum, a simple form with the possible choices on orientation is presented to the student. The student than communicates his or her interests to the agent through the form.

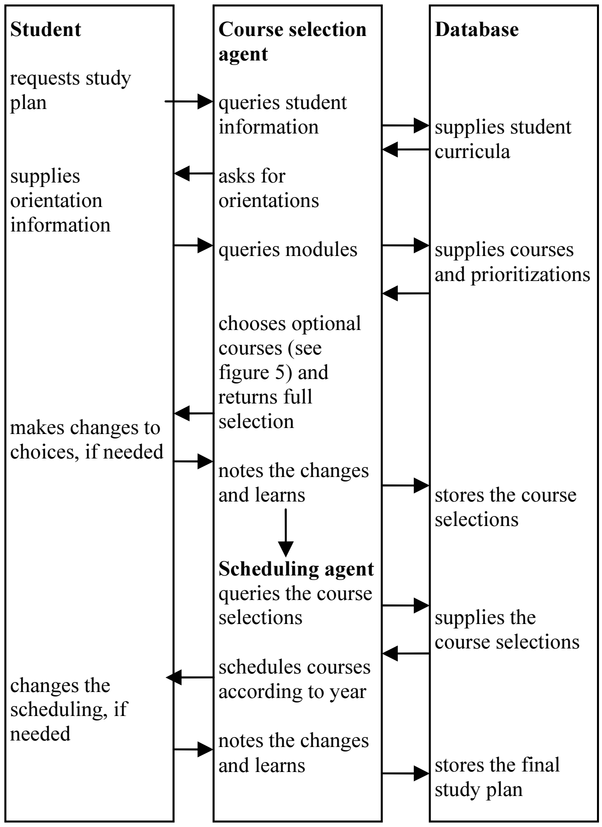

Figure 5: The process of the agent driven personal study plan

Based on that information, the agent queries the database for the information needed for choosing the courses. The courses chosen are then presented to the student, who can make changes as needed. The agent records the changes and learns from them. The agent also notes choices that were not changed so that it can base its future decisions on those.

After this, the agent stores the choices into the database and informs the scheduling agent so that it can begin its work. The scheduling agent queries the courses chosen by the student and also the learned information on which point of the students studies they fit best. Based on that information, the agent forms a full schedule on the courses, completing the study plan. The student can change any timing within the schedule. The agent will note those changes and learn accordingly. Finally, the study plan is stored in the database for further use.

2.6 Decision Making in the Agent System

The system has full information of the structure of the curriculum and from this information it can find the last modules which have optional courses in them. This is important as the approach used is that the student has a clear goal in his or her studies and the study plan is used to reach that goal.

Based on this approach, the choices on orientations made by the student are used to find the most suitable courses for the student based on learned suitability or values given by the expert.

After the courses have been chosen, the system uses the information on prerequisite studies and automatically adds them to its suggestion. After this, it moves down the curriculum to the next module with optional courses. At this point, the system may have already filled this module, if there were many prerequisites for the more advanced courses. If there is still a need for new courses, enough are chosen and the system checks for prerequisites again. This is repeated until the whole curriculum has been handled. In the end, all the mandatory courses are added.

With all the courses selected and after having given the student a chance to have his or her input on the choices by selecting or removing courses, the system can move onto arranging the courses by year.

A basic layout can be formed by using a topological sort. Many courses find their natural timing this way, but not all. The process continues by using the learned data on correct timings. The topological sort also gives the agent enough information to find a timing to suggest for new courses.

3 Data Storing Requirements for the Study Plan Environment

All of the information needed for the decision making and the study plans can be stored in a relational database. Figure 7 represents a possible structure for such database. This structure is used in the prototype version [5, 6].

The design is built around the student and the curricula. Curricula are divided into modules (basic studies, major, minor and so forth), which can have several different variants; although in many cases they don't have any. Modules are made up of courses and the student's study plan comprises of these with timing information.

As described in 2.4, the system requires some information on which courses are more advanced than others. This information is presented by forming a tree. The root is the curriculum itself and it usually has two children: bachelor level studies and master level studies. These are again divided into two or more sections and so forth. If a module has both mandatory and optional courses, these have been divided into different sections.

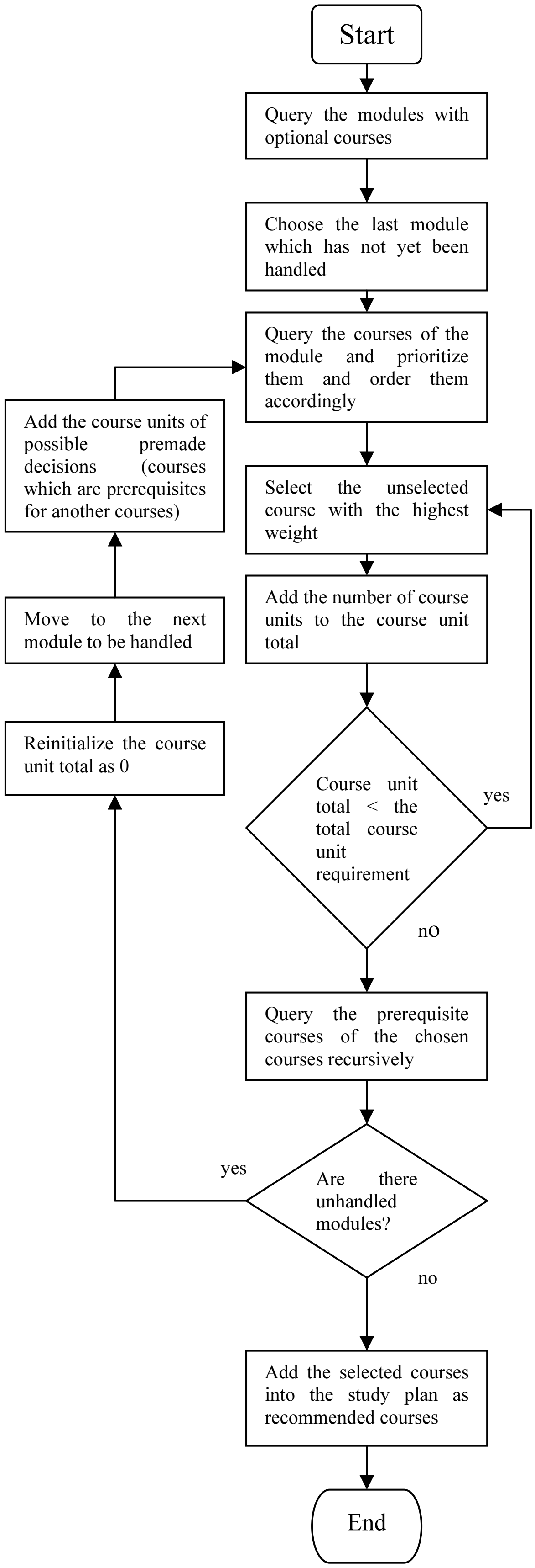

Figure 6. Algorithm for choosing the optional courses

The tree is represented in the database by giving each module the values ‘left’ and ‘right’. Beginning with the root, each vertex is given the values depth first. On the first visit, the ‘left’ value is assigned and on the final visit, the ‘right’ value is assigned. Thus root will have values of left is 1 and right is double the total number of vertices. The children of each vertex can be identified by using these numbers as both of the values of all the children will be between ‘left’ and ‘right’ of the given vertex. The main advantage of this method is the ease of depth first searches. The whole tree and thus the curriculum can be easily represented by ordering it based on ‘left’ values and using indentation [8].

Prerequisite courses are represented by a table with two separate foreign keys linking it to the courses. This table functions as the basis for forming the DAG [11] (see 2.2). An example of the data contained in a DAG is given in figure 6.

The orientation options are in their own table, with another table connecting them to the courses and the choices made on them by the students. The table connecting the orientations and courses holds the key information for the system to base its decisions on selecting courses. The table holds the weight proposed by the expert, who is working on the orientation, as well as the learned weight from previous choices by the students. Also, the number of students, from whom the weight was learned, is presented. This figure does not have to be accurate. It can be used to control the learning process somewhat. In the prototype, each course has a default 2 on this value and it can never go higher than 10, except momentarily.

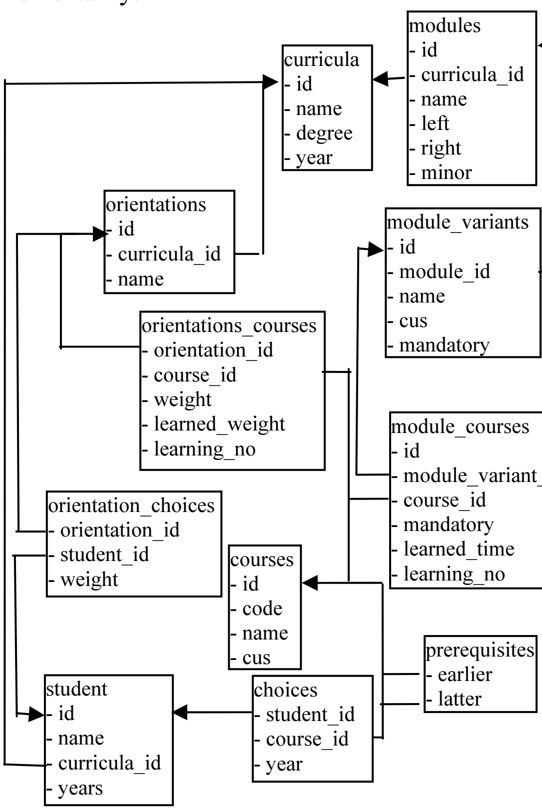

Figure 7: Structure of the database

The learned information on correct timing for courses is situated in the table ‘module_courses’ which represents which courses belong to which module.

The database structure is slightly redundant, as the information used for learning could be derived from the other tables, but the redundancy can be used for the aforementioned control and the structure can also make the system more efficient. The latter depends on how people use the system. If the study plan is constantly changed and not only looked over, the redundancy should probably be removed.

As the agent and decision approach proposed in section 2 is used, the classical Wompat-system of figure 1 is enhanced with an automated functionality on the personalized curriculum content selection.

4 Using Personal and Universal Information in Decisions of Agents

To keep the study plans up to date without the need for the student to check it regularly, the checking should be automated. This work can be done by an agent or several agents [9],

When the agent encounters a problematic choice, it notifies the student and begins to look for another course or courses which to suggest to the student. It can also leave a notification to a central message station that if others agents are having problems finding a suitable course they could perhaps make a common decision. This approach has the benefit of gathering a number of students to participate on a course so that the chance of that course being arranged is higher.

At this point, the agent must make a decision. Will it suggest another course, which is more interesting, but might not be arranged, if there aren’t enough students, or will it suggest a course which is probably arranged, but is not as interesting to the student as the other possibilities?

The agent could also make an attempt to make a deal with the other agents. If the problematic course is a key course in the students study plan, the agent could offer to change another course in its study plan if other agents are willing to change a course in their study plans to the key course. Student input is crucial at many of the stages, since the student has to actually carry out the plan.

Figure 8 represents one case where two agents might be able to guide their students into mutually beneficial agreement. Since neither can make decisions without receiving input from the students, the process is slow. Also, the process might result in nothing, if there still aren’t enough students who plan to take the course. The risk failure is increased by the fact that the students might get bored with the process.

In this case, the most important thing is to keep the student aware of the situation, so that the student can take action too if necessary or the agent can stop the negotiation if the course suggested is not to his or her liking.

Technically the decision making can be done with in the same way as before. The environment could be a ticket-like system built on the same database structure as the rest of the system. The agent can leave or read tickets from the database and use them for evaluation of its situation.

Agent 1 is looking at a study plan with four courses selected (**bold**) and four other possibilities. It also has access to the perceived suitability to the student in question (between 0 and 1). Agent 1 sees that not enough students have chosen course A, which is according to the figures very central in the study plan. Agent 2 is in a similar situation with course E.	**Agent 1** **- course A, 0.91** **- course B, 0.74** **- course C, 0.75** **- course D, 0.45** - course E, 0.44 - course F, 0.31 - course G, 0.22 - course H, 0.11	**Agent 2** - course A, 0.45 **- course B, 0.78** **- course C, 0.75** - course D, 0.44 **- course E, 0.94** **- course F, 0.46** - course G, 0.24 - course H, 0.31
Agent 1 finds that the next most suitable course would be E and agent 2 finds the course A (underlined). Just both making the change would help neither party. Also, if only one of the two makes a change, one is left without a key course. So, Agent 1 makes a suggestion. It is ready to give up course D in favor of E, if Agent 2 changes course F to A.	**Agent 1** **- course A, 0.91** **- course B, 0.74** **- course C, 0.75** **- course D, 0.45** - course E, 0.44 - course F, 0.31 - course G, 0.22 - course H, 0.11	**Agent 2** - course A, 0.45 **- course B, 0.78** **- course C, 0.75** - course D, 0.34 **- course E, 0.94** **- course F, 0.46** - course G, 0.24 - course H, 0.31
Since neither party has a strong bias between D and E or A and F, both might benefit if a change is made. The agents might need to persuade others to change their study plans too, but they have moved closer to their goal	**Agent 1** **- course A, 0.91** **- course B, 0.74** **- course C, 0.75** - course D, 0.45 **- course E, 0.44** - course F, 0.31 - course G, 0.22 - course H, 0.11	**Agent 2** **- course A, 0.45** **- course B, 0.78** **- course C, 0.75** - course D, 0.34 **- course E, 0.94** - course F, 0.46 - course G, 0.24 - course H, 0.31

Figure 8: Example of a negotiation between agents

Currently the problem of how to weight the risk of not being able to attend a course is unsolved. Perhaps this should be left to the student to decide or the system could learn to find the best way to go from experience. This learning might take years before it could reach a reasonable level of functionality and there isn't necessarily that much time.

5 Testing the Forming of the Study Plan

In the following tests the emphasis was on the learning capabilities of the system. As noted in section 2.4, learning is change in the system which results in better performance. Based on this, the test environment keeps records of changes made by the students. All changes can be regarded as wrong decisions by the system. If the number of those wrong decisions lowers over time, the system is able to learn. The tests were conducted with the software engineering students of University of Vaasa. First of the 20 cases was done by beforehand to give the system something to base its scheduling on. The other 19 were students ranging from first year to sixth year students.

In the beginning of the tests, it became obvious that the students were too willing to accept what the system chose as their courses. Only two of the students made any changes to the course selection and both only added courses to their selection. The small number of changes can probably be explained with the fact that the curriculum of these particular students has gone through very radical changes in the recent years. Because of this, the curriculum used for the tests might not have been the optimal choice.

student	course changes	schedule changes
1	0	39
2	0	21
3	0	21
4	4	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	26
10	0	14
11	0	15
12	0	8
13	2	7
14	0	8
15	0	11
16	0	9
17	0	9
18	0	4
19	0	7
20	0	5

Figure 9: Learning of the scheduling agent

The first two students both made 21 changes to their schedules. After that the system had adopted much of the information and next five students made no changes. Student number 9 chose a different amount of years for his study plan, which resulted in a need for 26 changes, which is the maximum of the series.

Because of student number 9, many of the learned values were radically changed, but the system seemed to be able to recuperate, although every student after that made changes. After long use, such radical changes should not happen. The progress of the learning can be seen in Figure 10:

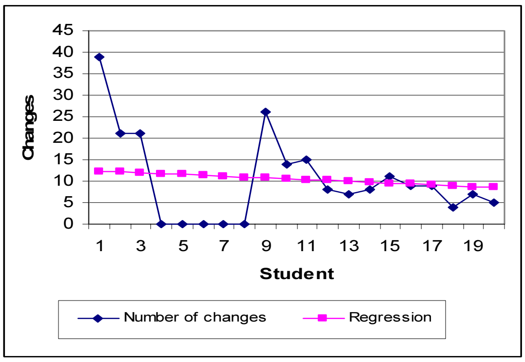

Figure 10: Learning of the scheduling agent represented as a graph

6 Conclusions

Finnish and EU students are going to need more guidance in the future. Obvious solution would be to hire more counselors, but as we have shown above, this is not the only option. In fact, our system could go beyond the capabilities of the counselor, if the whole system is used for decision making by the departments.

In this case, the system could be a usable tool for all parties: it can help the students plan their studies better which is also the goal of the public administration, and the departments can base their teaching plans on concrete and practical improvements.

The system is currently still under technical development [6]. The system can already identify the suitable courses and construct a timetable with good accuracy [5]. The solution is generic and can be used in a number of environments.

Acknowledgement

We wish to acknowledge all parties active in the development and testing of this system, all the people who have given their feedback on the system, the test subjects and especially the original developers of Wompat.

References

[1] Korkeakoulujen opintoaikojen lyhentämisen toimenpideohjelma (2003). Finnish Ministry of Education. In Finnish.

[2] OVI – ohjausta virtuaalisesti (2005). http://ovi.joensuu.fi. University of Joensuu. In Finnish.

[3] Oodi (2005). http://www.oodi.fi. University of Helsinki. In Finnish.

[4] Wompat-system. http://wompat.uwasa.fi. In Finnish.

[5] A. Vainio (2005). Opiskelijan opintojen suunnittelun avustaminen Wompat-järjestelmällä. In Finnish.

[6] Necora Systems Oy. http://www.necora.fi.

[7] S. Russell & P. Norvig (2003). Artificial Intelligence A Modern Approach, 2nd edition. Prentice Hall.

[8] J. Celko (2004). Trees and Hierarchies in SQL. Morgan Kaufmann Publishers.

[9] G. Weiss (ed.) (1999). Multiagent Systems A Modern Approach to Distributed Artificial Intelligence. The MIT Press.

[10] P. Mars, J.R. Chen & R. Nambiar (1996). Learning Algorithms – Theory and Applications in Signal Processing, Control and Communications. CRC Press.

[11] T. Cormen, C. Leiserson & R. Rivest (1990). Introduction to Algorithms. The MIT Press.

[12] The Bologna Process – Next stop Bergen. http://europa.eu.int/comm/education/policies/educ/bologna/bologna_en.html. 2005.