COMPUTER ASSISTED STUDY PLANNING

M. Pantic, A. Djordjevic, L.J.M. Rothkrantz, H. Koppelaar

Delft University of Technology
Faculty of Information Technology and Systems (ITS)
Zuidplantsoen 4, 2628 BZ Delft, the Netherlands
M.Pantic@cs.tudelft.nl.

KEYWORDS

Teleconsulting, Education using simulation and multimedia, Computer Based Training, Computer Assisted Instruction, Java

ABSTRACT

Automated Study Planning (ASP) system was designed to assist the students in organising their studies. Interactive expert system ASP forms a part of a bigger project that concerns computer-based education and training in higher learning [1]. The formulation of the study-planning process, whereupon the reasoning engine of ASP is built, has been acquired from the practical knowledge of the study-advisors of the faculty of Information Technology and Systems of TU Delft. The system is now on-line available via WWW pages of Knowledge Based Systems department TU Delft.

INTRODUCTION

In spite of the fact that computer technology, multimedia techniques and telecommunications are in the era of rapid development, approach to the process of education does not change much from the time of professors with periwigs. Students attend lectures in big auditoriums, practice in practical training rooms of the university and often live nearby their faculty. Life of a student is organised in a default way. A student cannot follow a lecture in his/her own tempo because the lecture is given to numerous of students in the same auditorium. A student cannot practice at the evening because the faculty building is already closed. Or simply, a student cannot take a day off because the lectures or the practical work will be missed.

Solution to this problem is "lecture-surfing" on WWW instead of lecture attending in a faculty auditorium. The World Wide Web, for much of its existence, has been a method for distributing various information to a widely distributed number of people. Tele-education via web will facilitate availability of lectures and practical assignments 24 hours a day from any corner of the world. Making tele-education a real possibility is an ongoing project of Knowledge Based Systems Department of the TU Delft. Tele-education will consists of three different systems integrated in a whole.

- CYBERCOURSE, containing general information, lectures, demonstrations and practical assignments of each subject of the faculty. The system represents a shell that can be easily emptied and filed again with information about different courses. It also contains so-called prick-board where the students can chat with each other or pose the questions to the teacher. The system is currently available via Web pages of TU Delft.
- ASP, representing an automated study-advisor. The system is available via WWW pages of TU Delft.
- AEA, representing an automated system for applying for the examination. The system will be read-only connected to the database containing information about students and their results and read-and-write connected to ASP and CYBERCOURSE. AEA represents in fact final integration and finalisation of Tele-education project but this isn't done yet. There is a reason to it. By connecting students-database with ASP and CYBERCOURSE the students will not be able to apply to the examination in the courses for which they do not have enough knowledge according to the study-advisor, what still does not mean that they cannot pass that exam. In other words, AEA will form, to a certain extent, a directed way of studying making the student unable to determine freely his/her own way. A decision hereabout is still to be made. Till then CYBERCOURSE and ASP will remain teleconsulting systems where given instructions/advice need not necessarily be followed.

AUTOMATED STUDY PLANNING

In a conventional organised study-advising system advice about the planning of the studies is given to the individual students personally by the study-advisors. Considering the fact that there are a lot of students which have a problem with organising their studies or which simply want to control the usefulness of their planning two major problems are encountered with the conventional study-advising system.

1. A lot of study-advisor's time is needed to help the individual students by giving quite routine advice about making a good study-planning.
2. Students cannot be individually advised as quick as they probably want to be.

Fig. 1: Conventional versus teleconsulting

As illustrated in *Figure 1*, these two problems of the conventional approach initiated the idea of automating the existent study-planning system. Implementation of this idea resulted in ASP. **Automated Study Planning** system represents a teleconsulting of the students in the matters of their study-planning and computer assisted training in making a good organisation of studies.

SYSTEM DESCRIPTION

The goal placed on the development of ASP is a fourfold.

1. **Teleconsulting** - the system must advise individual students in making a good study-planning in an
   - anonymous,
   - interactive and
   - user-friendly way.
2. **Computer assisted training** – the system must make the student aware of the necessity of a good study-planning and of the way of achieving it.
3. **Intelligent advisor** – the quality of the system's advising must be such that the study-advisors are relieved from advising the students about their study-planning.
4. **Universality** – the system must be easy modifiable to different studies and accessory courses.

LIFE CYCLE MODEL

The Linear Life Cycle Model [2] has been chosen for the development of the system. This life cycle consists of six different stages listed below and described through rest of this paper.

1. Planning Stage produced formal work plan for the development of the system [3]. During this stage the feasibility assessment [4] according to the checklist method [5] has been performed which resulted in several conclusions.

- Expert system is the appropriate paradigm for the solution of the given problem of an automatic study advisor because the task is primarily cognitive, complex, involves knowledge rather than numerical computation, requires reasoning and explanation of results while task knowledge is confined to a narrow domain and can be acquired from the experts (study-advisors) which are available and inspired [4][5][6].
- The payoff of the system is ensured considering that the study-advisors don't have to be as busy with individual study-planning of the students as without the system.
- Management commitment to fund the project and experts commitment to participate fully in the development of the system is ensured.

2. Knowledge Definition Stage.
3. Knowledge Design Stage.
4. Code and Checkout Stage.
5. Knowledge Verification Stage.
6. System Evaluation Stage.

KNOWLEDGE ACQUISITION

After the knowledge source has been identified (study-advisors) the process of the knowledge acquisition began. Two knowledge acquisition strategies [6] has been implemented:

1. structured interviews of study-advisors and
2. protocols, i.e. observing the study-advisors during the work-process.

The acquired knowledge has been divided into the several groups that form the knowledge base of the system.

- Number, names and lecture-year of the courses.
- Recommended and compulsory courses that have to be completed before starting with (applying for an examination in) a particular course.
- Priority of the courses according to the recommended and compulsory knowledge and to the difficulty of a particular course.
- Examination periods and dates.
- By the study-advisor as well as by the teacher recommended amount of the time necessary for preparation of a particular examine.
- Generally defined appropriate proportion between the number of exams and the amount of the available time.

Further, based on the acquired knowledge final functional layout, illustrated in *Figure 2*, has been generated. As illustrated, two main functions of the system have been determined.

1. **1.** Automatic Study Advisor (ASA) for processing the input of the student's personal data, results and choice of the study planning, generating an assessment of the planning and an advice about that planning, processing the input of the timetable data and generating an assessment and an advice on made timetable (see *Figure 3*).
2. **2.** Maintenance of Automatic Study Advisor (MASA) for maintenance of the data about study-advisors and of the validity of the passwords necessary to enter MASA function of ASP, maintenance of the data about courses and the related recommended and compulsory knowledge, maintenance of the data about examination periods and maintenance of the timetables of exams and their priority (see *Figure 4*).

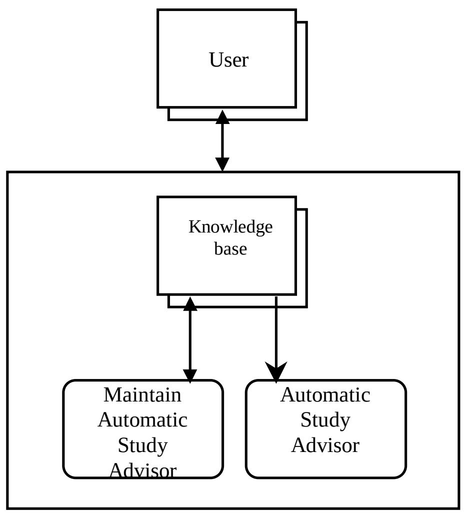

Fig. 2: DFD of ASP

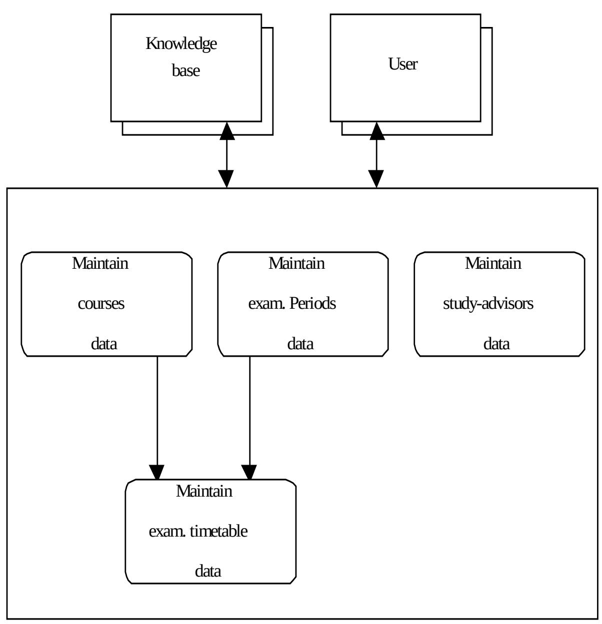

Fig. 4: DFD of MASA function

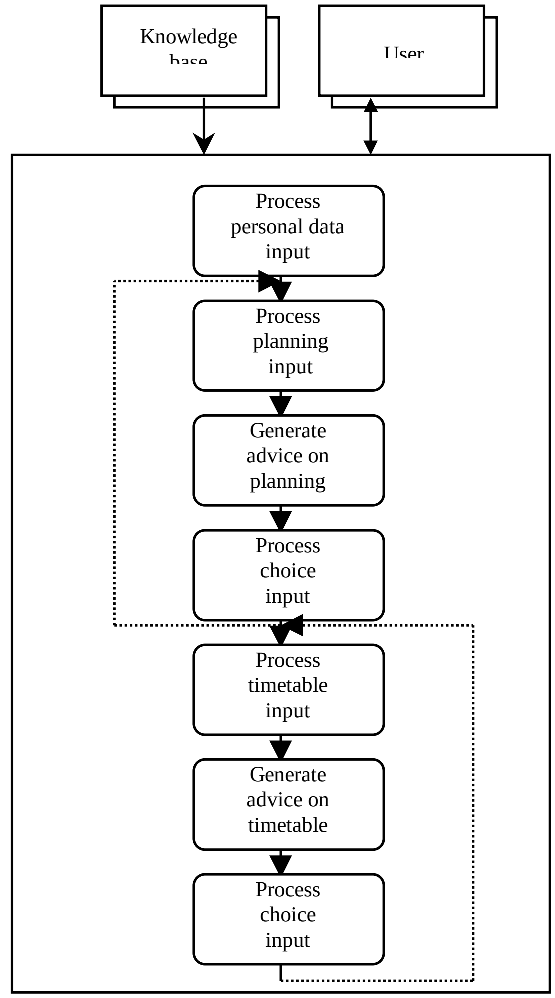

Fig. 3: DFD of ASA function

KNOWLEDGE DESIGN

The knowledge base has been connected with the inference engine of the system with two separate connections as illustrated in *Figure 2*. The first one is intended for the study-advisor only. This both-ways connection serves for changing the hard data of the knowledge base. This connection is protected from misuse through requiring of a valid password for establishing of the connection. The second, one-way connection is primarily intended for the students. The knowledge base is loaded from the web to the local machine of the student, so that a virtual data is formed which will be changed in accordance to the personal results and choices made by student.

Based on the information loaded by the student the inference engine reduces the virtual data present in the copy of the knowledge base and represents it to the student who makes a further choice. After loading his/her own results, making a choice of examination period and exams the student will get an assessment of his/her planning (see *Figure 3*). Further, the student gets the reasons for the assessment, which in fact form an overview of the made planning and the advice that can be followed in order to make a better one. An example of system's reasoning process can be represented in pseudo-code as illustrated in *Figure 5*. The outlook of the given advice about student's planning is illustrated in *Figure 6*.

```
IF
    (exam-A-is-selected) AND
    (NOT (exam-B-is-selected)) AND
    (exam-B-has-higher-priority-
        than-exam-A)
THEN
    apply-for-exam-B-It-has-
    higher-priority-than-exam-A
```

Fig. 5: Hypothetical example of ASP's reasoning

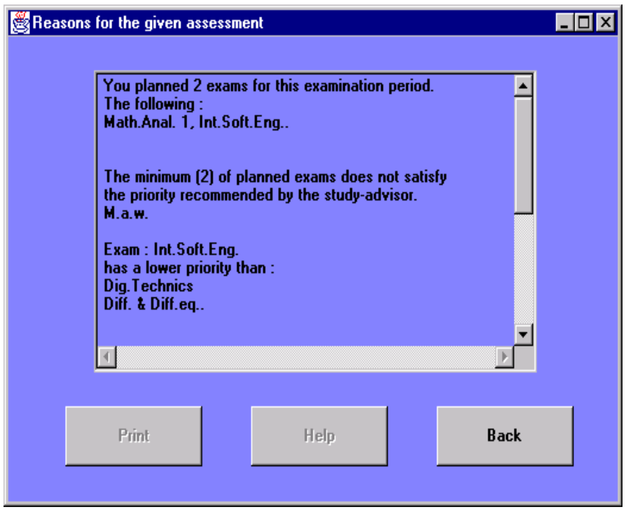

Fig. 6: Advising on made planning

After eventual changing of his/her planning according to the given advice, the student can proceed with making a timetable for preparation of the planned exams. In accordance to the available time, recommended amount of time necessary for the preparation of a particular exam and a-priori defined appropriate proportion between the number of exams and the amount of the available time, the system will make an assessment of the made timetable (see *Figure 3*). As in the case of the planning assessment, the reasons, representing an overview of made timetable as well as the advice that can be used to make a better timetable, are available to the student. The outlook of the given advice about student's timetable is similar to the outlook of the advice given about student's planning illustrated in *Figure 6*.

SYSTEM STRUCTURE

Functions of the system are implemented with a row of interactive windows, which form functional hierarchy. Term *functional hierarchy* specifies the relationships in the system. Two main functions consist of a number of sub-functions (see *Figure 3*, *Figure 4*). Each of the sub-functions is implemented separately (as one physical interactive unit) and further connected with its ascendant and descendant sub-function. The connections between the sub-functions are waterfall-like. In other words, to execute a particular sub-function of the system all of its ascendant sub-functions must be executed. For example, in order to generate an assessment of the study-planning it is necessary that the input of personal data, results and desired planning is already performed. Therefore it can be said that ASP has a structure determined by its functional hierarchy.

MAINTAINENANCE FUNCTION

One of the goals posed on the development of ASP was the universality of the system. The system must be easily modifiable to different faculties and the related courses. It is exactly MASA part of the system that makes it universal. On an interactive and user-friendly way the content of the knowledge base can be easily changed without affecting the mechanism or the resulting quality of the reasoning engine.

By interaction with MASA part of the system the study-advisors can step by step empty the existing knowledge base and fill it again with a new data that reflects the educational program of the faculty in question. By changing the contents of the knowledge base the system will constantly perform various controls so that no redundant or contradictory data occur in the knowledge base. An example of such control is given in *Figure 7*.

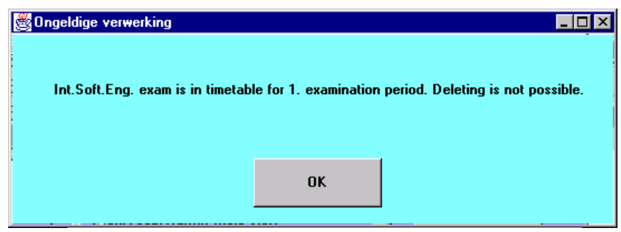

Fig. 7: Automatic avoiding contradictory

The outlook of the study-advisor's interaction with MASA part of the system is illustrated in *Figure 8*.

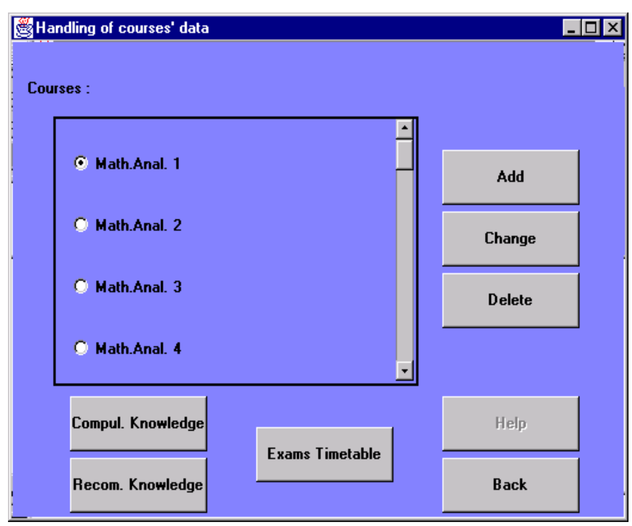

Fig. 8: Maintenance of courses' data

Probably the most satisfactory solution for the study-advisors would be achieved if they have nothing to do with the system once it is developed. Unfortunately, such a solution isn't really possible. Study-advisor is the only one who knows to determine correct priorities of the exams once one of the courses is changed or a new one is added to the educational program. Such changes as well as the changes in dates of exams must be handled once a year, when the new timetable of exams is made for the new school year.

It must be concluded that one data handling per year is not a high demanding maintenance.

SYSTEM IMPLEMENTATION

ASP is an Expert System. As already mentioned, when the study of feasibility has been performed it was concluded that Expert System is a good paradigm for implementing an automatic study advisor [4][5][6].

ASP is built as a Rule-Based Expert System. The reasoning mechanism of the system forms in fact a set of condition clauses (see *Figure 5*). The best way of implementing condition clauses is to implement them as rules.

ASP is implemented in Java. Enter Java and the capability for Web pages to contain Java applets, make interactive applications reality. Java applet is a dynamic and interactive program that can run inside a Web page displayed by a Java-capable browser [7]. Considering the goal of ASP project – teleconsulting in an anonymous, interactive and user-friendly way – it isn't so strange that ASP is implemented as Java applet.

KNOWLEDGE VERIFICATION

Before the system has been made available via WWW pages of TU Delft several tests has been performed. Each test has been analysed in terms of the incorrect-, incomplete- and inconsistent answers, and in terms of determination if the problem lies in rules, inference chain, or some combination of these two factors. The results of the tests have been documented and handled before it was proceed with further testing.

At the point when no more problems have been encountered by the developer two kinds of user-test has been committed.

- Testing by expert – The goal of this test was final verification of the built-in knowledge. Two from a total of three existing study-advisors on the faculty of ITS, have tested the knowledge elements as well as the rules of the system's reasoning mechanism for incorrectness, incompleteness and inconsistency in different cases. In accordance to the test's result it can be concluded that no fault in the reasoning process has been found so far and that the quality of the given advice is satisfactory.
- Testing by user – The objective of this test was establishing of the extent to which the posed goal has been fulfilled. A group of five students tested the ASA part of the system with different input data. Two study-advisors tested the MASA part of the system. The result of this test shows at the first place that the system is user-friendly. In the case of the ASA part of the system, the test's result also shows that the generated advice is understandable and helpful for a better making of the study-planning.

After the system has been made available via WWW an E-mail address has been placed so that the reactions on the processing and the quality of the consulting provided by ASP can be received. No reactions has been received so far.

EVALUATION

Considering the results of the performed user test it can be concluded that the **ASP is** a **teleconsulting system** that generates helpful advice about the study-planning and timetable of exams' preparation; a **computer assisted trainer** for enhancement of the time-organising skill; an **universal interactive, anonymous and user-friendly** tele-study-advisor.
Therefore it can be concluded that the goal posed on the development of ASP has been achieved.

A research toward estimation of the percentage of the students that still seek help about their study-planning from the study-advisor isn't officially undertaken. However an unofficial record reveals that during six mouths of system's existence 40% less students has asked for personal help from study-advisors then in the same period last year. Considering the fact that the system is still not well known, our opinion is that the above given unofficial record is quite promising.

REFERENCES

[1] Rothkrantz, L.J.M., van Vark, R.J., Bos, A.J.S.I.K. & Pauw, D.J.A. (1996). *An Educational Site on the World Wide Web*. Euromedia '96 proceedings.

[2] Bochsler, D.C. (1988). *A Project Management Approach to Expert System Applications*. ISA 88, pp.1458-1466.

[3] van der Lelij, K.J. (1996). *LIA for automated study-planning*. Internal Report. Delft University of Technology, Knowledge Based Systems Department, Delft, NL.

[4] Boullart, L. (1992). *A Spiral Model of Software Development and Enhancement.* Pergamon Press Ltd., Headington Hill Hall, Oxford, UK.

[5] Beckman, T.J. (1991). *Selecting Expert System Application*. AI Expert, pp.42-48.

[6] Steels, L. (1992). *Knowledge Systems.* Addison-Wesley Ltd., Amsterdam, NL.

[7] Flanagan, D. (1996). *Java in a Nutshell*. O'Reilly & Associates Inc., Sebastopol, CA, USA.