Almeida P., Novais P., Costa E., Rodrigues M., Neves J., Artificial Intelligence Tools for Student Learning Assessment in Professional Schools, in Proceedings of the 7th European Conference on e-Learning, Cyprus, November, ISBN 978-1-906638-22-1, pp 17-28, 2008.

Artificial Intelligence Tools for Student Learning Assessment in Professional Schools

Paulo Almeida[1], Paulo Novais[2], Eduardo Costa[1], Manuel Rodrigues[3] and José Neves[2]
[1]Centro de Formação Profissional da Indústria de Calçado, São João da Madeira, Portugal
[2]Departamento de Informática/CCTC, Universidade do Minho, Braga, Portugal
[3]Escola Secundária Martins Sarmento, Guimarães, Portugal
paulo@cfpic.pt
pjon@di.uminho.pt
director@cfpic.pt
MF.rodrigues@esmsarmento.com
jneves@di.uminho.pt

Abstract: The necessity to maximize the learning success of the students as well as to produce professionals with the right skills to fulfil the market requirements, raises the question of closely following and assessing the learning paths of the students of Professional Schools. To solve at once problems and difficulties that arise during the learning process, we need to develop technologies and tools that allow the monitoring of those paths, if not in real time, at least periodically.
Supported on a knowledge base of student features, also called a Student Model, a Student Assessment System must be able to produce diagnosis of student's learning paths. Given the wide range of students' learning experiences and behaviours, which implies a wide range of points and values in students' models, such a tool should have some sort of intelligence. Moreover, that tool must rely on a formal methodology for problem solving to estimate a measure of the quality-of-information that branches out from students' profiles, before trying to diagnose their learning problems.
Indeed, this paper presents an approach to design a Diagnosis Module for a Student Assessment System, which is, in fact, a reasoner, in the sense that, presented with a new problem description (a student outline) it produces a solved problem, i.e., a diagnostic of the student learning state.
We undertook the problem by selecting the attributes that are meaningful to produce a diagnosis, i.e., biographical, social, economical and cultural data, as well as skills so far achieved, which may drive, as constraints or invariants, the acquisition of new knowledge. Next, we selected the metrics that would allow us to infer the quality of the ongoing learning, i.e., the degree of expertise on the currently attended learning domains. To collect these indicators we used the Moodle e-Learning System. Both, attributes and metrics, make the student model. Finally, we designed a reasoner based on Artificial Intelligence techniques that rely on the Quality-of-Information quantification valuations to foster a Multi-Valued Extended Logic Programming language, a key element in order to produce diagnosis of the student learning paths. Confronted with a new case, i.e., a student model, the reasoner evaluates it in terms of its QI and outputs a diagnostic.

Keywords: Artificial Intelligence, Multi-Valued Extended Logic Programming, Rule Based Programming, Quality-of-Information, MOODLE.

1. Introduction

Professional Schools were born amid a view of an industrial society with the mission to qualify workers for different activity sectors or industrial clusters. Their student population was quite homogeneous in its social backgrounds and expectations, and the knowledge available was relatively narrow and stable. In this environment, it was easy to control the process of teaching and learning and to guarantee the success of the students, as well as to fulfil the entreprises' requirements of skills. However, nowadays, these schools are immersed in an information society, local and global at the same time, with a student population that comes from different social backgrounds, carrying with it different needs and expectations. Moreover, the enterprises, their traditional clients, are pressing the schools for more qualified technicians, able to help them embody the technological revolution to keep up with the global competition. Schools are, therefore, confronted with a new technological paradigm, a new kind of public and new demands from the enterprises.
Professional Schools have tried to cope with these challenges by investing in organization, management and market research, and in human and technological resources, as well as in new pedagogical tools, such as e-Learning platforms and, in some cases, Tutoring Systems. That way they hope to maximize student success along with student knowledge.

These investments are very expensive; schools can not afford to have unsuccessful students. As a consequence, the students' careers must be closely followed. Schools should have devices to evaluate their students' profiles, i.e., they should possess means to keep their students' profiles up to date, that way being able to follow and diagnose, in real time, the learning paths, to avoid failures as much as possible. Furthermore, the need to supply the market with effectively qualified personnel favours these evaluations.
The evaluation and following of the students' learning paths should be performed by teachers and psychologists, who access and diagnose the profiles and paths of the students to detect symptoms of deviations and act accordingly.

1.1 Overview of the Student Assessment System

Our aim is to design and develop a tool that could help these decision makers assess the students' learning processes. Its purpose is to help teachers and others to detect signs of learning problems on a particular student, and to take the proper measures to overcome them. In order to produce reliable results, the student's profile attributes must be evaluated in terms of its accuracy to the diagnosing process. The presence or absence of information must also be taken into account. In other words, the information quality of the set of attributes that compose the student profile must be measured before producing a diagnosis of the student learning path. Each diagnostic is labelled with its quality of information. If the teacher feels that the diagnostic is not reliable because of low quality of information, he/she can ask for more data or for more accurate data (by conducting questionnaires to the student, for instance). In addition, we need a language to express the knowledge about the student and, at the same time, to handle this uncertainty about the quality of the information.

1.2 Structure of this work

This work presents an approach to design a Student Assessment System (SAS), giving a strong emphasis on its diagnosis module and on the Quality-of-Information (QI) that stems out from the analysis of the student data. It is intended to produce a Decision Support System (DSS) to help teachers, tutors, psychologists and others, to detect in time problems on the students' evolution and decide and take the appropriate measures to work them out.
The SAS uses data from several sources to build and evolve student models. Based on those models it presents the decision makers with a diagnostic of the student learning course. We start by summarizing the notions of e-Learning Systems and present some features of MOODLE e-Learning System that are of interest to this work. Next we formalize the knowledge representation in SAS and evaluate the QI of a student model, as a way to assess the reliability of the acquired knowledge. Finally we describe the concept of student models and system beliefs, and our vision of a SAS, including its Student Model Builder and, mainly, its Student Diagnosis Module.

2. e-Learning Systems and MOODLE

e-Learning systems are software programs that provide support to learning activities. They include personal training systems, usually designed for a certain knowledge domain, known as Tutoring Systems (VanLehn, 2006: 227-265), as well as general learning management tools suitable to manage distinct types of learning content, covering several knowledge domains.
The Learning Management Systems/Course Management Systems (LMS/CMS) are domain independent, general purpose programmes/platforms, which provide authoring, sequencing, and aggregation tools that structure content to ease the learning process. It is the responsibility of the course designer to select and organize the matter in order to build a pedagogical module for a knowledge domain. MOODLE platform (Moodle - A Free, Open Source Course Management System for Online Learning, n.d.) is an example of a LMS/CMS. It is widely used and the number of installations is growing at a fast pace. It is also the platform that supports our study, though this study is generic enough to be applied to any other LMS.

The MOODLE Learning Management System

MOODLE (Modular Object-Oriented Dynamic Learning Environment) has a number of interactive learning activity components like forums, chats, quizzes and assignments. Very interesting is the lesson activity, wherein it is possible to write interactive learning tasks with conditional paths, adapting to the student learning process. In addition, MOODLE includes a logging module to track users' accesses and the activities and resources that have been accessed. Administrators and teachers can extract reports from this data. Figure 1 shows a high level view of the MOODLE modules.

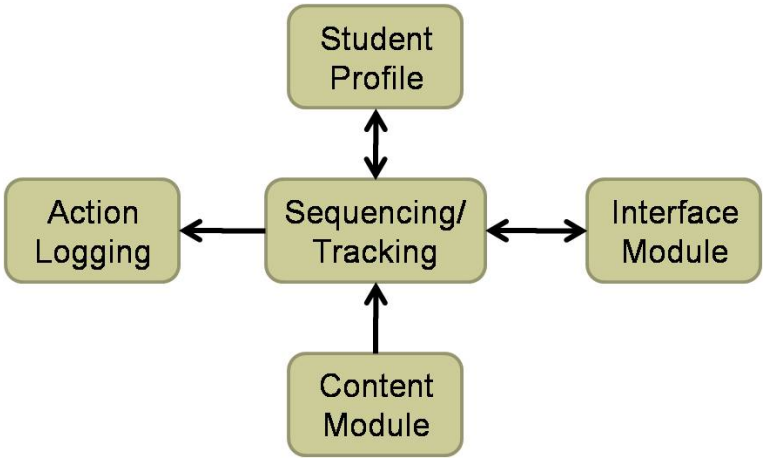

Figure 1: MOODLE LMS modules

Whatever the case, Tutoring Systems or LMSs, systems should have some sort of knowledge about the students and about their learning processes. This knowledge, i.e., the beliefs the system has about the students, is usually called the Student Model (SM). Without a SM a system would simply behave the same way for all students.

3. Knowledge representation for information quality assessment

Knowledge representation is a crucial factor regarding the success of the operation of a DSS (Neves, 1984; Way, 1991; Analide et al., 2006; Ginsberg, 1991).
A suitable representation of incomplete information and uncertainty is needed, one that supports non-monotonic reasoning.
In a classical logical theory, the proof of a theorem results in a *true* or *false* truth value, or in an *unknown* value. On contrary, in a Logic Program (LP), the answer to a question is only *true* or *false*. This is a consequence of the limitations of the knowledge representation in a LP, because explicit representation of negative information is not allowed. Additionally, the operational semantics applies the Closed-World Assumption (CWA) (Hustadt, 1994) to all predicates. Usually, LP represents implicitly negative information assuming the application of reasoning according to the CWA.
An extended logic program (Program 1), on the other hand, is a finite collection of rules of the form (Neves, 1984; Gelfond and Lifschitz, 1990):

$$q \leftarrow \wedge\, p_1 \wedge \ldots \wedge p_m \wedge \text{not } p_{m+1} \wedge \ldots \wedge \text{not } p_{m+n} \quad (1)$$

$$? \, p_1 \wedge \ldots \wedge p_m \wedge \text{not } p_{m+1} \wedge \ldots \wedge \text{not } p_{m+n} \quad (2)$$

Program 1: An Extended Logic Program

where *?* is a domain atom denoting falsity, p_i and q are classical ground literals, i.e., either positive atoms or atoms preceded by the classical negation sign ¬. Every program is associated with a set of abducibles. Abducibles can be seen as hypotheses that provide possible solutions or explanations for given queries, here in the form of exceptions to the extensions of the predicates that make the program.
The objective is to provide expressive power for representing explicitly negative information, as well as directly describe the CWA for some predicates, also known as predicate circumscription (Parsons, 1996: 353-372). Three types of answers to a given question are then possible: *true*, *false* and *unknown* and the representation of null values is scoped by Extended Logic Programming (ELP) (Analide et al., 2006)
Using ELP, as the logic programming language, a procedure given in terms of the extension of a predicate called *demo,* is given by Program 2. This predicate allows one to reason about the body of knowledge presented in a particular domain, set on the formalism referred to above. Given a question, it returns a solution based on a set of assumptions. This meta-predicate is defined as:

Demo: Question x Answer

where *Question* denotes a theorem to be proved and *Answer* denotes a truth value: *true* (T), *false* (F) or *unknown* (U).

$$\text{demo}(Q,T) \leftarrow Q$$
$$\text{demo}(Q,F) \leftarrow \neg Q$$
$$\text{demo}(Q,U) \leftarrow \text{not } Q \wedge \text{not } \neg Q$$

Program 2: An extension of the meta-predicate *demo*

4. Quality-of-Information of student models

We have seen that one may not always reason based only on LP representations of system beliefs. We have also seen how to use ELP to express uncertainty and overcome this limitation. In any decision making process, the decision is made without having all the information pertaining to the problem. When the decision maker has to, he/she makes the decision using the available information, to the best of his/her knowledge. How much a teacher relies on the diagnostics at hand? How can SM provide him/her with a measure of the quality of that information?

Let i ($i \in 1 \ldots m$) represent the predicates whose extensions make an extended logic program that models the universe of discourse; and j ($j \in 1 \ldots n$) the attributes of those predicates. Let $x_j \in [min_j, max_j]$ be a value for attribute j. To each predicate is also associated a scoring function $V_{ij}[min_j, max_j] \rightarrow 0 \ldots 1$, that gives the score that predicate i assigns to a value of attribute j in the range of its acceptable values, i.e., its domain (for simplicity, scores are kept in the interval [0 … 1]), here given in the form:

all(attribute_exception_list, sub_expression, invariants)

This denotes that *sub_expression* should hold for each combination of the exceptions of the extensions of the predicates that represent the attributes in the *attribute_exception_list* and the *invariants*.

This is further translated by introducing three new predicates. The first predicate creates a list of all possible exception combinations (pairs, triples, ..., n-tuples) as a list of sets determined by the domain size (and the invariants). The second predicate recourses through this list and makes a call to the third predicate for each exception combination. The third predicate denotes *sub_expression*, giving, for each predicate, the respective score function. The Quality-of-Information (QI) with respect to a generic predicate P is therefore given by $QI_P = 1/Card$, where *Card* denotes the cardinality of the exception set for P, if the exception set is not disjoint. If the exception set is disjoint, the quality of information is given by:

$$QI_P = \frac{1}{C_1^{Card} + \cdots + C_{Card}^{Card}}$$

where C_{Card}^{Card} is a card-combination subset, with *Card* elements.

The next element of the model to be considered is the relative importance that a predicate assigns to each of its attributes under observation: w_{ij} stands for the relevance of attribute j for predicate i (it is also assumed that the weights of all predicates are normalized, i.e.:

$$\forall i \sum_{j=1}^{n} w_{ij} = 1$$

It is now possible to define a predicate's scoring function, i.e., for a value $x = (x_1 \ldots x_n)$ in the multi-dimensional space defined by the attributes domains, which is given in the form:

$$V_i(x) = \sum_{j=1}^{n} w_{ij} * V_{ij}(x_j)$$

And it is possible to measure the QI that occurs as a result of a logic program that makes a SM, by posting the $V_i(x)$ values into a multi-dimensional space and projecting it onto a two-dimensional one. Using this procedure, it is defined a circle, as the one given in Figure 2.

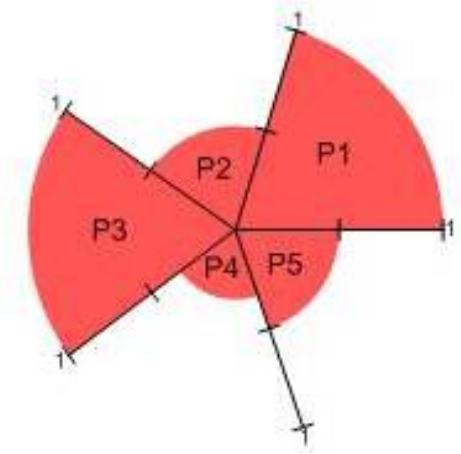

Figure 2: A measure of the QI for a Logic Program or Theory P

Here, the dashed n-slices of the circle (in this example built on the extensions of five predicates, named as p_1 ... p_5) denotes the QI that is associated with each of the predicate extensions that make the logic program. Now, we can evaluate the QI of the SMs of Ana and Diana. Let us consider the logic Programs 3 and 4, which represent a set of beliefs about students, as well as exceptions to those beliefs.

$$\begin{aligned}
&\text{school_att}(\text{ana}, 1)\\
&\text{had_attended}(\text{ana}, \text{geometry})\\
&\neg\text{perc_right_answers}(S, L, V) \leftarrow \text{not perc_right_answers}(S, L, V),\\
&\qquad\qquad \text{not exception}(\text{perc_right_answers}(S, L, V))\\
&\text{exception}(\text{perc_right_answers}(\text{ana}, \text{shoecad_SW}, 80))\\
&\text{exception}(\text{perc_right_answers}(\text{ana}, \text{shoecad_SW}, 50))
\end{aligned}$$

Program 3: An example of the Universe of Discourse for Ana SM

$$\begin{aligned}
&\neg\text{school_att}(S, V) \leftarrow \text{not school_att}(S, V), \text{not exception}(\text{school_att}(S, V))\\
&\text{exception}(\text{school_att}(S, V)) \leftarrow \text{school_att}(S, \bot)\\
&\neg\text{had_attended}(S, V) \leftarrow \text{not had_attended}(S, V), \text{not exception}(\text{had_attended}(S, V))\\
&\text{exception}(\text{had_attended}(S, V)) \leftarrow \text{had_attended}(S, \bot)\\
&\neg\text{perc_right_answers}(S, L, V) \leftarrow \text{not perc_right_answers}(S, L, V),\\
&\qquad\qquad \text{not exception}(\text{perc_right_answers}(S, L, V))\\
&\text{school_att}(\text{diana}, \bot)\\
&\text{had_attended}(\text{diana}, \bot)\\
&\text{exception}(\text{perc_right_answers}(\text{diana}, \text{shoecad_SW}, 12))\\
&\text{exception}(\text{perc_right_answers}(\text{diana}, \text{shoecad_SW}, 40))
\end{aligned}$$

Program 4: An example of the Universe of Discourse for Diana SM

QI associated with students Ana and Diana is depicted in Figures 3 and 4, respectively.

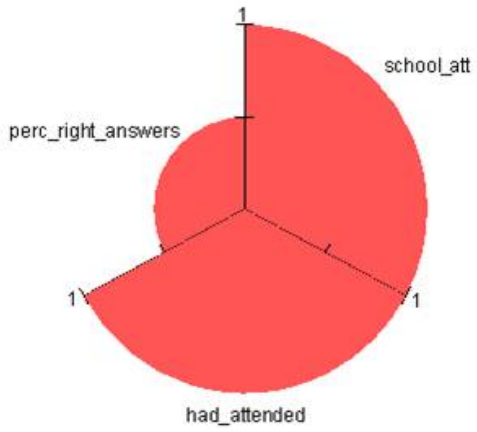

Figure 3: A measure of the Quality-of-Information of Ana SM

In order to find the relationships among the extensions of these predicates, we evaluate the relevance of the QI, given in the form $V_{school\_att}(ana)= 1$; $V_{perc\_right\_answers}(ana)= 0.5$; $V_{had\_attended}(ana)= 1$. Roughly, this means we are sure about school attitude and attendance information of Ana; but we are not so sure about the information on percent of right answers. As for Diana, we are not sure whatsoever about her school attitude and attendance, although we have some assurance (0.5) about her percent of right answers.

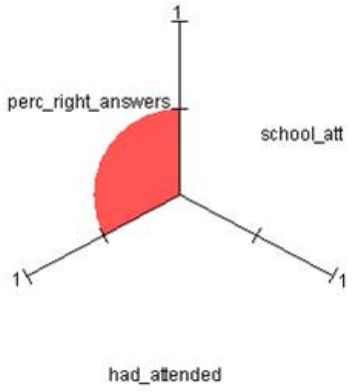

Figure 4: A measure of the Quality-of-Information of Diana SM

5. Student Models

Therefore, a SM may be defined as a representation of the set of beliefs that a system has about a student (Self, 1994). We will follow Self's definition but we will use Multi-Valued Extended Logic Programs (MVELP) to express those beliefs, being the truth values bound to a proven theorem given in terms of a composition of the measures of the Quality-of-Information of the predicates that make it (Neves, et al 2007). Indeed, let $B_A p$ denote that a program A subscribes the substance (i.e., the essence of the extension) of predicate p. The belief set B_A configures the extensions of the set of predicates assumed by program A: $B_A= \{p \mid B_A p\}$. By applying this line of thought, we can define:

B_S as the student set of beliefs;

B_C as the computer system set of beliefs. This set includes the extensions of those predicates that the system believes with reference to the general student behaviour; and

SM as the subset of the system's beliefs which are beliefs that the system C has about the student S: $B_C(S)= \{p \mid B_C p(S)\}$.

The student's beliefs are not known by the system; therefore, all reasoning about the student has to be made through the analysis of the SM. As for B_C, this set contains the domain knowledge, beliefs

about student behaviour, as well as SMs, among others sources of information. The relationship between these sets is shown in Figure 5.

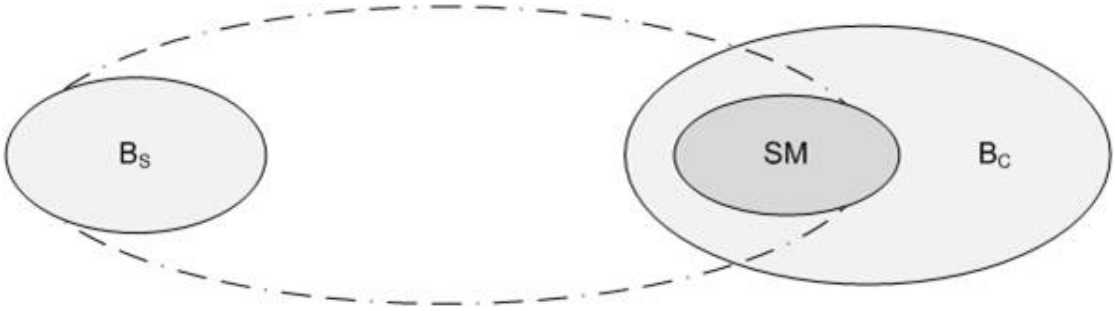

Figure 5: Relationship between B_S, B_C and SM (Self, 1994)

MOODLE does not have a true SM. It only has a student profile. However, MOODLE does collect metrics about all sorts of actions made by the users. SMs can be built from the history logs of the platform and updated with student activity logs, as we shall see later. In fact, there is a great amount of discussion about the feasibility of SMs from student interactions with learning platforms. Arroyo et al. (2004: 782-784), Johnson et al. (2005) and Mislevy et al. (1999: 437-446), just to name a few, have some work done on this area, mainly through the use of Bayesian Networks in conjunction with Data Mining, to model students' behaviour.

6. Student Assessment System

A SAS configures a DSS made of two modules: a module to create and update the SM; and a module to diagnose the learning path of the student and update the beliefs of the system, B_C. The former is the Student Model Builder (SMB); the latter is the Student Diagnosis Module (SDM).
SMB and SDM may be viewed as agents operating concurrently on the SAS environment. SAS is indeed a Multi-Agent System (Wooldridge and Jennings, 1995: 115–152; Jennings, 1999: 1-7; Jennings 2001: 35-41), whose architecture is depicted in Figure 6.

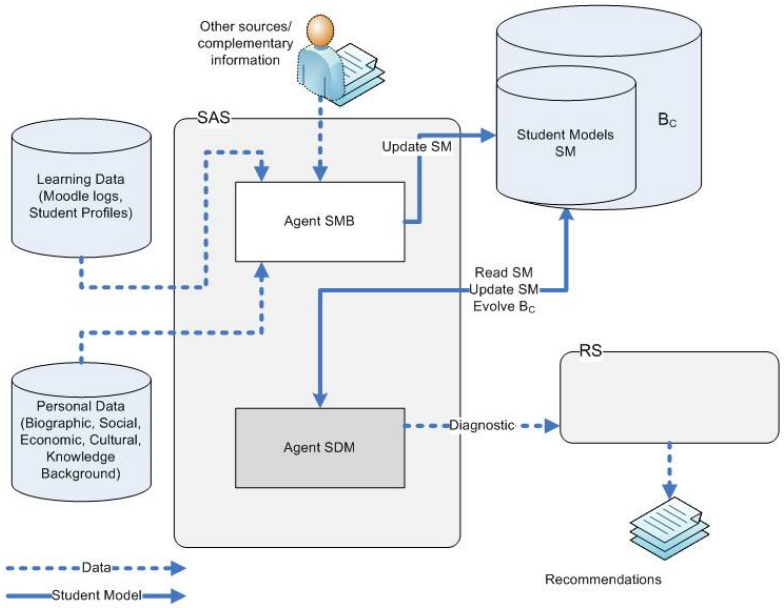

Figure 6: The Student Assessment System Architecture

7. Building system beliefs and student models from MOODLE

A SM does not have to be very complex and fully accurate to be useful. “The computational effort to improve correctness may not justify the extra pedagogical leverage obtained. Computational utility, not cognitive fidelity, is the measure for student models” (Self, 1994).
In order to be able to produce SMs, the computer system must have some beliefs about student behaviour, face to the pedagogical resources from a given domain. These beliefs, as we had already seen, are integrated into B_C.
MOODLE has an activity logger to register users’ accesses (i.e., user ID, IP and time of access) and the activities and resources that have been accessed. From the log, MOODLE is able to generate, for each student, activity reports. This information can be combined with biographical, socio-economic and cultural data, as well as former academic history, in order to obtain a more complete representation of the students’ evolution. Figure 7 presents a simplified information model of MOODLE activity log.

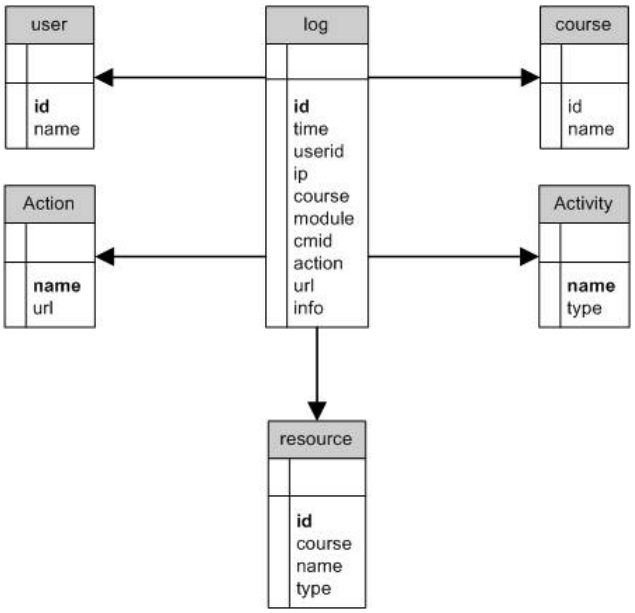

Figure 7: Excerpt from MOODLE Information Model

7.1 Extracting system beliefs B_C

Applying a methodology for mining association rules from databases, similar to those used by Lukichev, Diaconescu and Giurca (2007: 437-446), which are in itself based on the works of Agrawal, Imielinski and Swami (1993: 207–216) and Agrawal and Srikant (1994: 487–499), it is possible to derive a set of beliefs from MOODLE logs of student transactions to establish B_C. Generally, the idea is to find patterns, correlations and associations on data sets of student interactions with MOODLE. The discovering of association rules is used to come across elements that occur together in data sets with a given confidence and support. Confidence and support, parameters defined by the user, may be used to establish an order relation on the beliefs (however, in this work, such parameters will not be object of consideration, for the sake of simplicity).
We use Rule Based Programming (RBP) (Kowalski and Levy, 1996), here given in terms of productions of the Multi-Valued Extended Logic Programming (MVELP) language. In RBP, given a rule in the form $Q \leftarrow P$, Q is triggered whenever P occurs. As an example, we may express association rules in terms of a MVELP (Program 5):

$$B_C = \begin{cases} \ldots \\ \text{lesson\_success}(S, \text{shoecad\_SW}) \leftarrow \text{had\_attended}(S, \text{geometry}) \\ \text{lesson\_success}(S, \text{shoecad\_SW}) \leftarrow \text{num\_lesson\_visits}(S, \text{shoecad\_SW}, N) \wedge N >= 20 \wedge \\ \qquad \text{avg\_lesson\_time}(S, \text{shoecad\_SW}, T) \wedge T >= 900 \\ \text{good\_learning\_att}(S, \text{shoecad\_SW}) \leftarrow \text{num\_lesson\_visits}(S, \text{shoecad\_SW}, N) \wedge N >= 15 \wedge \\ \qquad \text{avg\_lesson\_time}(S, \text{shoecad\_SW}, T) \wedge T > 600 \\ \text{grade}(S, \text{shoe\_design}) \leftarrow \text{lesson\_success}(S, \text{shoecad\_SW}) \wedge \text{school\_att}(S, 1) \\ \ldots \end{cases}$$

Program 5: Extract of system beliefs B_C

In this example, the system believes:

- That those who had attended *geometry* lessons will probably be successful on lesson *shoecad_SW*;
- That high number of visits to lesson *shoecad_SW* and high average time spent with its pedagogical resources, usually predicts success on that lesson;
- That some number of visits to the lesson with some average lesson visit time probably means that the student is having a good learning attitude;
- That those who succeed on *shoecad_SW* and have good school attitude usually grade on *shoe_design* course.

Systems beliefs can also be added or edited manually by teachers, tutors, psychologists or others.

7.2 Building and updating student models

The SM can be initialized through a combination of system default assumptions and question forms presented to the students by the time of their enrolment on a course, being updated through the student interactions with the system. Besides, there is information we can not derive from log files (e.g., school attitude or socio-economical status). Indeed, this information must come from other sources.

Let us consider the following subset of MOODLE metrics, for a given student, attending a given course:

num_lesson_visits: number of visits to a lesson;
avg_lesson_time: average time spent per lesson; and
perc_right_answers: percent of right answers.

For example, two students, Ana and Diana, attending a *shoe_design* course, studying *shoecad_SW* lesson, may have the following SMs (Figure 8).

$$SM_{Ana} = \begin{cases} \text{school\_att}(\text{ana}, 1), \text{had\_attended}(\text{ana}, \text{geometry}) \\ \text{num\_lesson\_visits}(\text{ana}, \text{shoecad\_SW}, 22) \\ \text{avg\_lesson\_time}(\text{ana}, \text{shoecad\_SW}, 950) \\ \text{perc\_right\_answers}(\text{ana}, \text{shoecad\_SW}, 60) \end{cases}$$

$$SM_{Diana} = \begin{cases} \text{school\_att}(\text{diana}, 0), \text{NOT had\_attended}(\text{diana}, \text{geometry}) \\ \text{num\_lesson\_visits}(\text{diana}, \text{shoecad\_SW}, 6) \\ \text{avg\_lesson\_time}(\text{diana}, \text{shoecad\_SW}, 200) \\ \text{perc\_right\_answers}(\text{diana}, \text{shoecad\_SW}, 12) \end{cases}$$

Figure 8: Programs for Ana and Diana SMs

These logic programs stand for the facts that the system knows about these students. The system beliefs about student behaviour have not yet been applied to the SMs of Ana and Diana. In fact, there is another source of information from which the student model may be updated, i.e., the current content of the system beliefs, B_C.

8. The Student Diagnosis Module

As we have seen, once the student model has been initialized, there are two sources of information on the basis of which the student model may be updated: the student's inputs to MOODLE, and the current contents of B_C.

The role of the SDM is to output a diagnostic and update the SM with beliefs about the student. Generally, the SDM gets the SM of a given student, analyses the SM, evaluates the quality of SM's information, outputs a diagnostic and updates the SM with its beliefs about the student. These beliefs result from instantiating the system set of beliefs, B_C, to this particular student. Figure 9, below, stands for the SDM architecture.

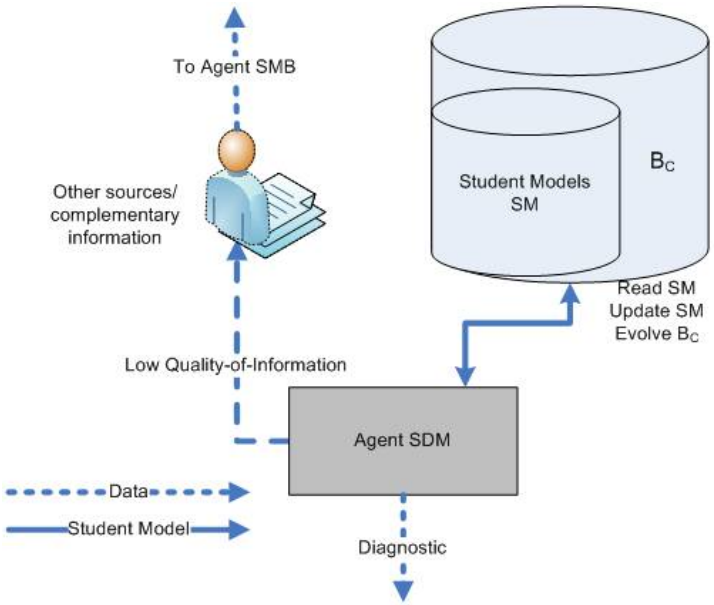

Figure 9: Student Diagnosis Module Architecture

For instance, it is possible to produce these diagnostics for Ana and Diana (Figure 10), as instances of the system beliefs B_C given in Program 8 above.

$$SM_{Ana} = \begin{cases} school\_att(ana,1), had\_attended(ana,geometry) \\ num\_lesson\_visits(ana,shoecad\_SW,22) \\ avg\_lesson\_time(ana,shoecad\_SW,950) \\ perc\_right\_answers(ana,shoecad\_SW,60) \\ lesson\_success(ana,shoecad\_SW) \\ good\_learning\_att(ana,shoecad\_SW) \\ grade(ana,shoe\_design) \end{cases}$$

$$SM_{Diana} = \begin{cases} school\_att(diana,0), NOT\ had\_attended(diana,geometry) \\ num\_lesson\_visits(diana,shoecad\_SW,6) \\ avg\_lesson\_time(diana,shoecad\_SW,200) \\ perc\_right\_answers(diana,shoecad\_SW,12) \\ NOT\ lesson\_success(diana,shoecad\_SW), \\ NOT\ good\_learning\_att(diana,shoecad\_SW), \\ NOT\ grade(diana,shoe\_design) \end{cases}$$

Figure 10: SMs after applying the system beliefs, B_C

Based on its convictions, the system believes Diana has problems: she will not succeed on *shoecad_SW* lesson, she has not a good learning attitude, and will not grade on course *shoe_design*. With this diagnostic, it is expected that teachers and others will help Diana correcting her learning path. Lying on the contrary, the system believes Ana will succeed.
We can see that the system updated the SMs with instantiations of its beliefs about *lesson_success*, *good_learning_att* and *grade*. These beliefs stand for the information given by SDM. The diagnostics are labelled with the QI of the SMs under scope.

When the SDM encounters evidence that the current SM is inaccurate, for example, by observing that the student acts differently (e.g., higher number of visits to a course, more time spent with resources, higher percent of right answers) to the way the SM would predict, then the SDM may try to find and adjust those components necessary to enable the model to correspond to the observed behaviour. This has implicit a time dimension not considered here.

9. Conclusions

The particular nature of the learning process demands grounded decisions taken in time, to be able to detect problems when they arise and avoid failures. A SAS, as the one we have described, is able to build and evolve SMs from the logs of an e-Learning platform. It is also able to output diagnostics of student learning paths with a certain Quality-of-Information. The Quality-of-Information is critical to assure the credibility of the whole process of diagnosing and, therefore, the decisions of the decision makers. There is too much at stake when we deal with learning processes, and we must raise the confidence on decisions, especially in an environment of uncertainty, incomplete and imperfect information.
The framework just presented is a theoretical model of a SAS. The development is at an early stage. For now, in order to achieve a first implementation, we favour the simplicity of the model. After the first version and the analysis of its results, we aim to evolve to a Recommendation System, Eventually, the output data of the SAS, i.e., the diagnostics, could be stored in a knowledge base together with the proposed therapies and a measure of success of those therapies, Based on this knowledge, the Recommendation System could be able to suggest therapies to overcome the problems diagnosed. It should have a reasoning mechanism like Case-Based Reasoning: confronted with a new diagnostic, this system could recommend a successful therapy formerly suggested for a diagnostic similar to the one at hand. However, at this moment, the role of the Recommendation System is still left to teachers, tutors and psychologists.

Acknowledgements

We would like to thanks the Professional School *Centro de Formação Profissional da Indústria de Calçado, São João da Madeira, Portugal*, for the kindness in letting us use its MOODLE e-Learning Platform in our studies and research.

References

Agrawal R., Imielinski T., Swami A. (1993) “Mining association rules between sets of items in large databases”, Buneman and Jajodia, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 207–216, Washington, D.C., USA
Agrawal R., Srikant R. (1994) ‘Fast algorithms for mining association rules’, Bocca, Jarke, and Zaniolo, editors, *Procedures of the 20th International Conference on Very Large Data Bases, VLDB*, pp. 487–499, Morgan Kaufmann Publishers, Los Altos, California, USA.
Almeida, P. (2008) *Tools for Student Learning Assessment in Professional Schools*, Technical Report, University of Minho, Braga, Portugal.
Analide C., Novais P., Machado J., Neves J. (2006) "Quality of Knowledge in Virtual Entities“, *Encyclopaedia of Communities of Practice in Information and Knowledge Management*, Coakes and Clarke, editors, Idea Group Reference, pp. 436-442.
Arroyo, I., Murray, T., Woolf, B. and Beal C. (2004) “Inferring unobservable learning variables from students’ help seeking behaviour”, *Workshop Proceedings of International Conference on Intelligent Tutoring Systems*, pp. 782-784.

Gelfond, M. and Lifschitz V. (1990) “Logic Programs with Classical Negation”, in *Proceedings of the International Conference on Logic Programming*.

Ginsberg, M.L. (1991) *Readings in Non-monotonic Reasoning*, Morgan Kauffman Publishers, Los Altos, California, EUA.

Hustadt, U. (1994) “Do we need the closed-world assumption in knowledge representation?” in *Working Notes of the KI'94 Workshop*, Saarbrüken, Germany.

Jennings, N.R. (1999) “Agent-Oriented Software Engineering”, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering*, Vol. 1647, pp.1-7.

Jennings, N.R. (2001) ”An agent-based approach for building complex software systems”, *Communications of the ACM*, Vol. 44, No. 4, pp. 35-41.

Jonsson, A., Johns, J., Mehranian, H., Arroyo, I., Woolf, B., Barto, A., Fisher, D. and Mahadevan, S. (2005) “Evaluating the feasibility of learning student models from data”, *Proceedings of the Workshop on Educational Data Mining at Association for the Advancement of Artificial Intelligence (AAAI)*.

Kowalski, T. and Levy, L. (1996) *Rule-Based Programming*, Springer.

Lukichev S., Diaconescu M. and Giurca A. (2007) ”Empowering Moodle with Rules and Semantics”, *Proceedings of the European Semantic Web Conference*, Innsbruck, Austria, May 30.

Mislevy, R.J., Almond, R.G., Yan, D., and Steinberg, L.S. (1999) “Bayes nets in educational assessment: Where do the numbers come from?”, Laskey and Prade, editors*, Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 437-446.

Moodle - *A Free, Open Source Course Management System for Online Learning*, [Online], Available: http://moodle.org/ [10 Jul 2008].

Neves, J. (1984) “A Logic Interpreter to Handle Time and Negation in Logic Data Bases”, in *Proceedings of the ACM'84, The Fifth Generation Challenge*.

Neves, J., Machado, J., Analide, C., Abelha, A., and Brito, L. (2007) “The Halt Condition in Genetic Programming”, in *Proceedings of the 13th Portuguese Conference on Artificial Intelligence, EPIA*, Guimarães, Portugal, pp. 160-169.

Parsons, S. (1996) ”Current approaches to handling imperfect information in data and knowledge bases”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 3, pp. 353-372.

Self, J. (1994) *Formal Approaches to Student Modelling*, AAI/AI-ED Technical Report No. 92.

VanLehn K. (2006) “The behaviour of tutoring systems”, *International Journal of Artificial Intelligence in Education*, Vol. 16, No. 3, pp. 227-265.

Way, E.C. (1991) *Knowledge Representation and Metaphor*, Kluwer Academic Publishers, Dordrecht, Holland.

Wooldridge, M. and Jennings, N.R. (1995) “Intelligent Agents: Theory and Practice”, *Knowledge Engineering Review*, Vol. 10, No. 2, pp. 115–152.