Enhancing user support in open problem solving environments through Bayesian Network inference techniques

N. Tselios*#, A. Stoica*, M. Maragoudakis+, N. Avouris*, V. Komis#
University of Patras, Greece
* Human-Computer Interaction Group, Electrical and Computer Engineering Department,
# ICT in Education Group, Dep. of Educational Sciences and Early Childhood Education
+ Wireless Communication Laboratory, Electrical and Computer Engineering Department
nitse@ee.upatras.gr, stoica@ee.upatras.gr, mmarag@wc.ee.upatras.gr, avouris@ee.upatras.gr, komis@upatras.gr

Abstract – During the last years, development of open learning environments that support effectively their users has been a challenge for the research community of educational technologies. The open interactive nature of these environments results in users experiencing difficulties in coping with the plethora of available functions, especially during their initial efforts to use the system. In addition, -from the tutors' perspective- the problem solving strategies of the students are often particularly difficult to identify. In this paper, we argue that such problems could be tackled using machine learning techniques such as Bayesian Networks. We show how we can take advantage of log files obtained during field studies to build an adaptive help system providing the most useful support to the student, according to the state of interaction. On the other hand, we attempt to support the tutor, by automating the process of diagnosing students' problem solving strategies using Bayesian Networks. The presented approaches are discussed through examples of two prototypes that have been developed and corresponding evaluation studies. These studies have shown that the proposed approach can effectively support the tasks of students and tutors in such open learning environments.

Keywords: Bayesian Belief Networks, Open problem solving environments, Inference algorithms, On-line adaptation, Adaptive help, Automated problem solving strategy identification.

Introduction

During the last years, a number of open problem-solving environments have been built that are based on the constructivist approach. Open problem solving environments are computer-based learning environments that let students actively explore certain concepts while they are engaged in problem solving, with emphasis in the active, subjective and constructive character of learning (Luger and Stubblefield, 1998). Compared to traditional learning environments the student's activity cannot be reduced in a sequence of pre-defined tasks. In general, through an open problem solving environment the student is engaged in processes such as model building, investigation and reflection. As a result, students have the freedom to explore various entities in their own unique way. The open nature of these environments, results in users' activity taking various forms and the patterns of use of the tools that are included in them not to be fully anticipated. Therefore, the complexity of such environments and the open nature of the tasks can often lead to poor usability, which can be an obstacle to obtain the expected pedagogical value from their use.

In general, through an open problem solving environment the student is engaged in processes such as model building, investigation and reflection. The process of building such a model is usually carried out in three phases. First, the system allows users to express their ideas, through 'entities' that are related to objects, corresponding to their phenomenological status. These objects have properties, directly manipulated by the students, and behaviour. At the second level, the students can relate explicitly or implicitly a number of entities to create a more abstract entity, considered also as a 'construct', depicting an object from a group of uniform objects, that take meaning in the context of a phenomenon, system, process or conjecture (Komis et al., 2001, Dimitracopoulou and Komis, 2005). Finally, the students are able to 'run' the model to validate their hypothesis and observe the function of their model, often aided by data visualisation tools such as equation plots, graphs, etc (Komis et al., 2002).

Improving usability of these environments is an objective that could be achieved in various ways. By applying user centred design techniques (Norman, 1986) at design time or by building user support components for run time. User centred design propose advanced usability evaluation techniques during system design. For instance, various usability evaluation techniques that take into account both the pedagogical value and the usability of the environment have been proposed (Squires and Preece, 1999, Avouris et al., 2000). Furthermore, more complex user task modelling approaches have been proposed for the design of such environments (Tselios et al., 2002). On the other hand, user support at run time through adaptive systems is an approach that could improve usability. It should be recognized that, overall, artificial intelligence techniques have not succeeded to deliver the expected results in the educational field in the form of Intelligent Tutoring Systems, despite the existence of some sporadic promising results (Woolf et al., 2001, see also Kinshuk and Russell, 2002, for a classification of adaptable and adaptive systems). However, the premise of adaptive system behaviour through which higher usability and increased system transparency can be obtained remains a valid scientific objective. Our approach is compatible with the vision of Suthers et al. (2001) which stress the value of 'Minimalist AI' in education: Instead of trying to build smart machines that teach -a rather optimistic goal demanding a great effort of modeling knowledge in a particular subject, as well as pedagogy strategies and explanation mechanisms-, this approach suggests providing machines with abilities to 'respond to the semantics of student activities and constructions, test the educational value of these abilities, and add functionality as needed to address deficiencies in the utility of the system'.

In the research reported in this paper, we attempt to investigate the usefulness of Bayesian Networks in tackling two significant problems, common in open learning environments. These two problems are complementary in nature and address difficulties of students and tutors when engaged in activities in open learning environments:

- First, from a *student perspective*, an adaptive help module is presented which automatically recognizes the most probable next action and presents relevant help items. The need for such an adaptive help system is significant, especially in the case of an open problem solving environment where the cognitive effort to deal with the task is affected by concerns on how to handle the various tools that may be used for accomplishing a certain task.
- From a *tutor perspective*, we attempt to present a practical method to classify effectively problem solving strategies expressed by pupils while they were using such an open problem solving environment. Due to the nature of such environments, problem solving strategies could be numerous. Posteriori analysis of log data in order to identify the strategy expressed by the students could be a rather tedious and painstaking process. Thus, a method to automatically classify the solutions presented to the users could substantially increase the evaluation of the learning process.

In the presented cases, we try to implement a *state-based* AI approach instead of a rich *knowledge-based* (Nathan, 1998). We attempt to identify important states, features and repeatable patterns in the students' –system interaction cycle in order to infer about their intentions or adopted strategy during their problem solving activity, while retaining at the same time the sense of self-activity and engagement. Bayesian networks are used as modelling tools in this state based approach in both cases.

This paper is organized as follows: First, we present an overview of interesting applications of Bayesian Belief Networks (BBN) in various problems in the field of educational technology. Then, we briefly present Bayesian probabilistic theory together with a description of the notion of BBN's and their implementation in educational environments. Next, we present our approach to build an adaptive help system in the *ModelsCreator* open problem solving environment. The proposed approach of adapting user support through a BBN is described. The BBN was built using a large amount of log files of actual usage of the system. Next, a system testing and evaluation experiment that took place in order to demonstrate the benefits of the proposed architecture is presented, followed by a short presentation of the results obtained from our experience of student modelling with BBN in open problem solving environments. Finally, we present our approach to automatically identify users' strategies during their interactions with an open learning environment that is used for teaching geometrical concepts to pupils of 11-14 age. In the proposed approach, a BBN has been used to infer problem-solving strategies that the pupils applied in order to solve a given problem from logs of activity of typical pupils. In this case, identification of user's strategies was done with the aim to facilitate classification of the solutions presented by the students and aid the evaluation of the learning process, a typical tutor task.

Use of Bayesian Belief Networks in Educational Systems

In recent years, symbolic modelling approaches applied in traditional AI problems, have been replaced by implicit models built from rich data sets through techniques proposed by Machine Learning and Knowledge Discovery from typical data of a specific domain. These techniques have produced efficient algorithms during the last years and found new areas of application. Among them, a technique that has certain advantages and has received attention during the last years is based on Bayesian Belief Networks (BBN's, Niedermayer, 1998). A Bayesian Belief Network (Stephenson, 2000) is a directed acyclic graph, where each node represents a random variable of interest and each edge represents direct correlations between the variables.

Bayesian reasoning is based in formal probability theory and is used extensively in several current areas of research, including pattern recognition, decision support and classification. Assuming a random sampling of events, Bayesian theory supports the calculation of more complex probabilities from previously known results (Luger and Stubblefield,

1998). The advantages of BBNs include the simple process for constructing probabilistic networks even from relatively limited amount of data, the efficient algorithms to evaluate degrees of belief for instances of a node and the versatile knowledge representation which such networks provide.
Due to the underlying probabilistic model that describes the belief on the existence of a specific event, BBNs are considered one of the most effective ways to represent uncertainty. By formulating a limited, approximate but representative cognitive model of the users, suitable for the specific problem, and then modeling uncertainty in human computer interaction, future activity could be supported. This is done through modeling of user's activity, recorded in click streams of typical user behavior when interacting with a software system. This could lead to interpretation of the nature of the cognitive processes involved and to more efficient ways to support future users' tasks. This approach is particular suitable for learning applications, in which interactions are complex and support is often needed.

So in recent years, various prototypes have been produced, demonstrating that BBNs are suitable for effective modelling of student behaviour (Jameson et al., 1995). BBNs have been utilized in various ways to achieve adaptability in educational environments in terms of determining student goals, determining feedback, curriculum sequencing and fine-tuning of the pedagogical strategy to deliver knowledge. ANDES (Conati et al., 1997), which is a system that teaches physics problem solving techniques to college students, uses BBNs to identify the current problem solving approach of the user. ANDES also use a BBN to determine what hints to provide to the user by identifying how the student is solving a problem and how he has progressed down the solution path. Extensions of the modeling process to cope with issues that arose in scaling up the model to a full-scale, field evaluated application coupled with results of several evaluations of Andes which provide evidence on the accuracy of the implemented models are presented by Conati et al., (2002). Bunt and Conati (2003) attempt a generalization to the student modelling process in open problem environments. They argue in favour of using such a technique, founding it beneficial even for those learners who do not already possess the skills relevant to effective exploration. The scope of their model is twofold: To assess the effectiveness of a learner's exploratory behaviour in an open learning environment and to monitor the learners' actions in the environment unobtrusively in order to maintain the unrestricted nature of the interaction that is one of the key assets of such environments.

In addition to reasoning at a lower interaction level about student actions it is possible to use a BBN in order to reason at a higher level. Collins et al. (1996) constructed a BBN that represents a hierarchy of skills for arithmetic, containing the test questions, the theory and the user's skill in specific topics. The semantics implied by the links represent which question refers to a specific topic. The system considers the current estimate of the student's ability to compute the probability of responding correctly to each question contained in the database schema. Then the system selects an item that the student has a near to 50% chance of answering correctly. Regarding the selected pedagogical approach, the criteria of item selection could be altered appropriately. CAPIT, (Mayo and Mitrovic, 2000), is a constraint-based tutor for English capitalization and punctuation that uses BBN for long term student modeling. An evaluation of this system showed that a group of students using the adaptive version learned the domain rules at a faster rate than the group that used the non-normative version of the same system.

Usage of Bayesian Networks is not limited to monitor the student's behaviour in an educational environment. For example, Vomlel (2004) stresses the use of Bayesian Networks to educational testing. He presented a series of case studies referring to operations that use fractions. He showed that a Bayesian network that models relations between required skills to carry out these operations, improve the process of student's evaluation. Also Xenos (2004), presents a methodological approach based on Bayesian Networks for modelling the behaviour of the students of a bachelor course in computers in an Open University that deploys distance educational methods. This method offered an effective way to model past experience, which can significantly aid decision-making regarding the educational procedure. According to Xenos (2004) this approach can also be used for assessment purposes regarding current state enabling tutors to identify best and worst practices.

As discussed in this section, probabilistic models, such as BBN have focused on traditional "drill and test" systems in great extend. However, in the research reported here, we attempt to show that applications of this method are not necessarily constrained only in the aforementioned type of systems. In the following, we show applications of BBNs in open learning environments with much more complex and non linear nature of user system interaction. As described in the rest of the paper, this class of educational systems could also benefit from a BBN approach, in a variety of ways.

Bayesian Modeling

A Bayesian Belief Network (BBN) is a significant knowledge representation and reasoning tool, under conditions of uncertainty (Mitchell, 1997). Given a set of variables D = $<X_1, X_2 \dots X_N>$, where each variable X_i could take values from a set $Val(X_i)$, a BBN describes the probability distribution over this set of variables. We use capital letters as X,Y to denote variables and lower case letters as x,y to denote values taken by these variables. Formally, a BBN is an annotated directed acyclic graph (DAG) that encodes a joint probability distribution. We denote a network *B* as a pair B=<G,Θ>, (Pearl, 1988) where *G* is a DAG whose nodes symbolize the variables of *D*, and *Θ* refers to the set of parameters that quantifies the network. *G* embeds the following conditional independence assumption:

Each variable X_i is independent of its non-descendants given its parents.

Θ includes information about the probability distribution of a value x_i of a variable X_i, given the values of its immediate predecessors. The unique joint probability distribution over $<X_1, X_2 \ldots X_N>$ that a network *B* describes can be computed using:

$$P_B(X_1 \ldots X_N) = \prod_{i=1}^{N} P(x_i \mid parents\ (X_i))$$

Learning BBN from data

The process of efficiently detecting the most suitable network is not straightforward. Thus, a BBN should be learned from the training data provided. Learning a BBN unifies two processes: learning the graphical structure and learning the parameters Θ for that structure. In order to seek out the optimal parameters for a given corpus of complete data, we directly use the empirical conditional frequencies extracted from the data (Cooper and Herskovits, 1992). The selection of the variables that will constitute the data set is of great significance, since the number of possible networks that could describe N variables equals to:

$$2^{\frac{N(N-1)}{2}}$$

where *N* is the number of variables (Jeffreys, 1939).We use the following equation along with Bayes theorem to determine the relation *r* (or Bayes factor) of two candidate networks B_1 and B_2 respectively:

$$r = \frac{P(B_1 \mid D)}{P(B_2 \mid D)} \quad (1) \qquad P(B \mid D) = \frac{P(D \mid B)P(B)}{P(D)} \quad (2)$$

where:

- *P(B|D)* is the probability of a network *B* given data *D*.
- *P(D|B)* is the probability the network gives to data D.
- *P(D)* is the ‘general’ probability of data.
- *P(B)* is the probability of the network before seen the data.

Applying equation (1) to (2), we get:

$$r = \frac{P(D \mid B_1)P(B_1)}{P(D \mid B_2)P(B_2)} \quad (3)$$

Having not seen the data, no prior knowledge is obtainable and thus no straightforward method of computing $P(B_1)$ and $P(B_2)$ is feasible. A common way to deal with this is to assume that every network has the same probability with all the others, so equation (3) becomes:

$$r = \frac{P(D \mid B_1)}{P(D \mid B_2)}$$

The probability the model gives to the data can be extracted using the following formula (Glymour and Cooper, 1999):

$$P(D \mid B)=\prod_{i=1}^{n}\prod_{j=1}^{qi}\frac{\Gamma(\frac{\Xi}{q_i})}{\Gamma(\frac{\Xi}{q_i}+N_{ij})}\prod_{k=1}^{ri}\frac{\Gamma(\frac{\Xi}{r_i q_i}+N_{ijk})}{\Gamma(\frac{\Xi}{r_i q_i})}$$

where:

- Γ is the gamma function.
- n equals to the number of variables.
- r_i denotes the number of values in *i:th* variable.
- q_i denotes the number of possible different value combinations the parent variables can take.
- N_{ij} depicts the number of rows in data that have *j:th* value combinations for parents of *i:th* variable.
- N_{ijk} corresponds to the number of rows that have *k:th* value for the *i:th* variable and which also have *j:th* value combinations for parents of *i:th* variable.
- Ξ is the equivalent sample size, a parameter that determines how readily we change our beliefs about the quantitative nature of dependencies when we see the data. In our study, we follow a simple choice inspired by Jeffreys (1939) prior. Ξ is equal to the average number of values variables have, divided by 2.

Given the great number of possible networks produced by the learning process, a search algorithm has to be applied. We follow greedy search with one modification: instead of comparing all candidate networks, we consider investigating the set that resembles the current best model most.

In general, a BBN is capable of computing the probability distribution for any partial subset of variables, given the values or distributions of any subset of the remaining variables. Note that the values have to be discretised, and different discretisation size affects the network. As we shall discuss in the result section, BBNs constitute a significant tool for knowledge representation, visualising the relationships between features and subsets of them. This fact has a significant result on identifying which features are actually affect the class variable, thus reducing training data size without any significant impact in the performance.

However, we should stress the fact that transforming the Bayesian Belief Network elicitation process to a tractable procedure is not enough. The most important step, while attempting to resolve a problem using application of such a technique, is the establishment of a deep understanding of the nature of the problem. This understanding could lead to a manageable transformation of the problem domain, where applications of BBNs could serve as an important inference tool. From our expertise obtained dealing with the adaptation problems described in the following, an abstract high level approach, attempting to re-express the problem to be modelled should include first an initial understanding and clarification of the prominent variables of the domain which influence our reasoning about the state or the goal of the student while executing a task. These variables, coupled with nodes indicating the issue (the ‘class’ variable) that is needed to infer upon, are used to feed the BBN construction process. The states of the variables represent actual user’s actions, neither in an high task level, nor in the lowest keystroke level of interaction. The granularity of the abstraction is heavily depended by the nature of the problem and our aim. It seems that different problems may need slightly different approach. For example, while attempting to recognise student’s current goal, the temporal order of her recent actions influences significantly our inference about her forthcoming action. When we attempt to infer general goals of the user from log files, it seems that focusing upon the frequency of tool’s usage is a preferable selection.

Using BBN to Construct an Adaptive Help System

This section presents an adaptive help system that was built using BBNs, for *ModelsCreator* (MC), an open learning environment for modelling activities of young students. ModelsCreator allows users to create models representing aspects and phenomena of the natural world and provides them with tools for testing their models (simulations, graphs, tables of values, etc.). *ModelsCreator* design and implementation has been inspired by two main design principles: (a) support of expression through different kinds of reasoning in a simplified and synthetic mode and (b) model mechanisms that derive from different subject matters, which permit interdisciplinary approaches (Komis et al., 2001, Dimitrakopoulou and Komis, 2005). Models can be built in MC, using qualitative reasoning, as well as quantitative and semi-quantitative relations that can be used to link various objects, representing primitive concepts. The system supports high level of visualisation, combines modeling tools with simulations and incorporates alternative and multiple forms of representation (Von Glasersfeld, 1987). Visual representations of entities in a specific topic (i.e. food, water, photosynthesis) constitute essential elements of the environment since they have mediating role of providing students with a ‘concrete’ framework to reason on an abstract level (Figure 1).

.

Concerning the MC user interaction design, a direct manipulation space, suitable for young students has been designed. A view of this space is shown in Figure 1, in which the modelling environment of MC can be seen. It has been shown (Komis et al., 2002, Ergazaki et al., 2005) that this system can provide a rich and constructive learning experience to young students. For example, in Figure 1, in the main modelling space a typical model of a biological phenomenon (photosynthesis) is shown. Entities like a plant, the sun, the soil, the leaf of the plant and the substances of the air included in the model and are related through qualitative relations. On the left, a library of entities is included. These can be dragged in the modelling space. On the right, the available relations are shown. Finally, on the top, various tools for model manipulation, for creating various representations of the model and running the model are included.

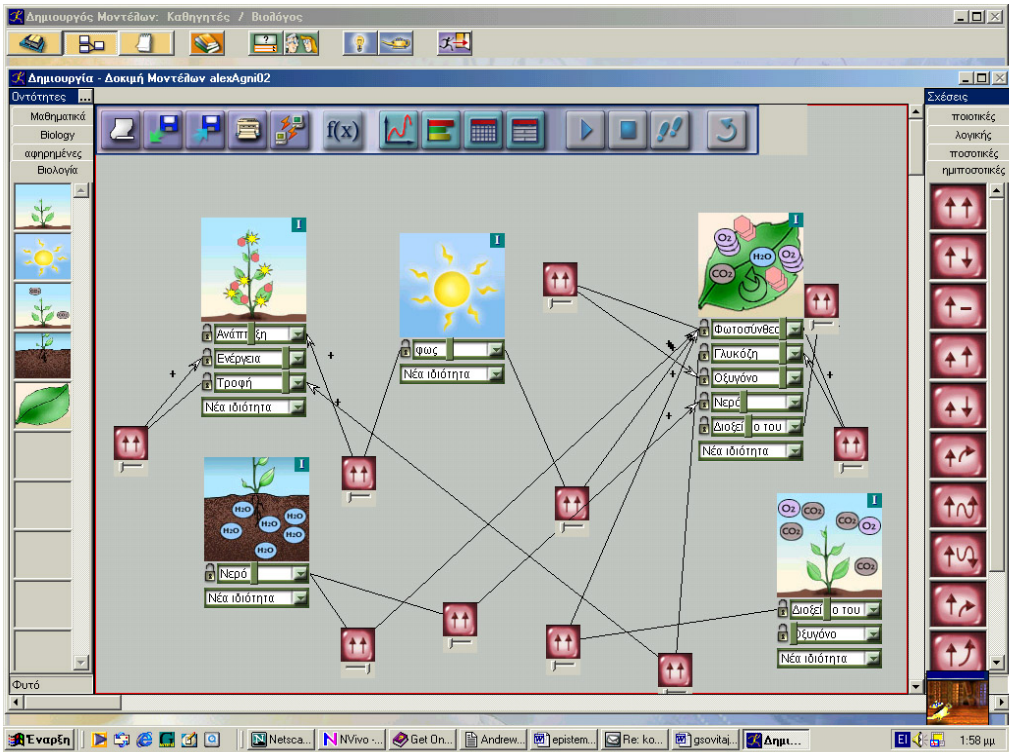

Figure 1. Models Creator’s modelling environment.

Online help adaptation using Bayesian networks

ModelsCreator contains a rich help system. However, it was discovered that in its earlier form this help system was particularly difficult to be used by young students, who had to spend long time browsing in large amounts of information in order to receive support. So a BBN has been developed in the context of the reported research, used in order to adapt the user support (help) system of ModelsCreator. The BBN was built using data collected from real world typical user interaction with the MC environment. The model was based on the selected data, so the approach we followed can be considered as a data-centric one. The data used to train the structure and the probabilities of the Bayesian network was obtained from a quite large amount of log files produced by ModelsCreator during previous experiments, involving 15 students of 11-14 years age group. Representative tasks were given to the students to accomplish. An extract of the log files used can be seen in Figure 2. This is a history of a sequence of operations of a user in the MC environment. Each operation might have associated attributes. So the first operation of the user was an insertion of an entity in the modelling space. The attribute of this specific action was the kind of entity that was inserted (the “Plant” entity).

A log file pre-processing procedure took place and resulted in building of a database containing some 2200 logged actions. Our effort was to construct a BBN in order to obtain a belief about the next most probable action of the student with respect to the current state of the interaction flow. We assumed that by predicting the next most probable operation, we can infer the user intention and thus relate the most relevant section of the help system if user sought support. The database that was built in this pre-processing phase contained the required fields to predict the next action given the two previous actions. The field *Property of the action* has been used for each action describing the particular nature of the action. For example, when a user selects a particular attribute of an Object, the selected attribute is recorded in the property field. The fields of the database are: *Previous Action* (Paction), *Previous Property of the carried out Action* (Pprop), *Previous Action Time*, *Current Action* (Caction), *Current Action Property* (Cproperties), *Current Action Time*, *Next Action* (Naction), *Next Property* and the *Time difference* between current and previous (CPTime_d). The database has this structure since the temporal ordering of the actions has not been taken into consideration.

The user activity in MC is very unpredictable. Therefore, there is virtually no constraint in solutions that the user might adopt and express during problem solving. Thus, an adaptation procedure in a higher cognitive goal level (e.g., concerning specific goals which the user attempts to accomplish, or even a sub-goal as a part of a pupil's approach to solve a problem) could be proved very difficult to be carried out. Instead, the proposed approach focuses on general patterns that can occur in the stream of user interaction with the software.

The log file extract shows a representative student's interaction with the system (Gudzial, 1993). First the user attempts to enter three entities (Plant, Leaf, Sun) to the MC's working space (Actions 1,3,4). Subsequently, she selects specific attributes for each entity, during her effort to build a model describing causal-effect relationships between the members of the model (Actions 2 and 5-12). Finally she executes the model, in order to observe the behaviour of the model. To further examine the exact nature of the relationships between the entities, she creates a barchart representing the evolution of the three selected entity's attributes over time (Actions 13-18).

#	**Time**	**Action**	**Property of the action**
01.	00 : 08 : 14	InsertEntity	Plant
02.	00 : 08 : 19	ChooseAttribute	Growth->Plant
03.	00 : 08 : 37	InsertEntity	Leaf
04.	00 : 08 : 40	InsertEntity	Sun
05.	00 : 08 : 46	ChooseAttribute	Photosynthesis->Leaf
06.	00 : 08 : 49	ChooseAttribute	Intensity of Light->Sun
07.	00 : 09 : 07	InsertRelation	Proportional
08.	00 : 09 : 07	ConnectRelation	Photosynthesis
09.	00 : 09 : 07	ConnectRelation	Intensity of Light
10.	00 : 09 : 48	InsertRelation	Proportional
11.	00 : 09 : 48	ConnectRelation	Growth
12.	00 : 09 : 48	ConnectRelation	Photosynthesis
13.	00 : 10 : 00	RunModel	
14.	00 : 11 : 19	BarChartActivation	
15.	00 : 11 : 21	GraphChooseAttribute	Growth->Plant
16.	00 : 11 : 23	GraphChooseAttribute	Intensity of Light->Sun
17.	00 : 11 : 24	GraphChooseAttribute	Photosynthesis->Leaf
18.	00 : 11 : 31	RunModel	

Figure 2. Extract of the log file used to build the BBN.

In the derived BBN (Figure 3), causal relationships between *Previous action* (PAction) and *Current action* (CAction), and between *Current action* and *Next action* have been depicted. From an interaction flow perspective, the structure is semantically meaningful and the relation between the previous and current action of a student is an expectable empirical model of interaction to find the probability of occurrence of an instance of the next action variable, only the current instance of the current action is needed. The structure of our Bayesian network is a tree structure denoting that it is suitable to be used even in real time training and adaptation as well, since the time required to obtain such a structure from data is polynomial (Stephenson, 2000).

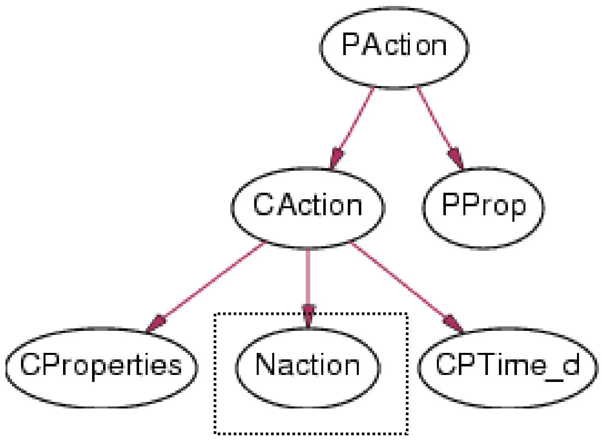

Figure 3. The final Bayesian network obtained. Previous action (PAction), Current action (Caction) and the time difference between them (CPTime_d), influences network's belief concerning the most probable next action.

Environment architecture modification

The architecture of Models Creator was enriched with a new module that incorporated the developed BBN and enhances the functionality of the User Support Module (Help System). The architecture of the new module named Adaptive User Support Module (AUSM) is conceptually presented in Figure 4.

The role of the new module is twofold. First, to collect data from the user's stream of actions and to write them into a database. Subsequently, to identify current interaction state using the collected data and to provide at any time the most appropriate help topics according to the probabilistic estimation made using the developed BBN. The database obtained is used as input for a Bayesian modelling tool that will process the data in the database and will construct the BBN that models our problem. The resulting network is exported in *Hugin Lite* file format. The module parses the file that contains the network and uses the conditional probability table within to provide the most useful help topics to ModelsCreator by predicting the most probable actions of the student to follow. This Bayesian network is actually used by a module that interacts with our open problem-solving environment, ModelsCreator and with a Bayesian modelling tool through files (it reads the BBN description file and it writes the database with the data to update the probabilities of the network).

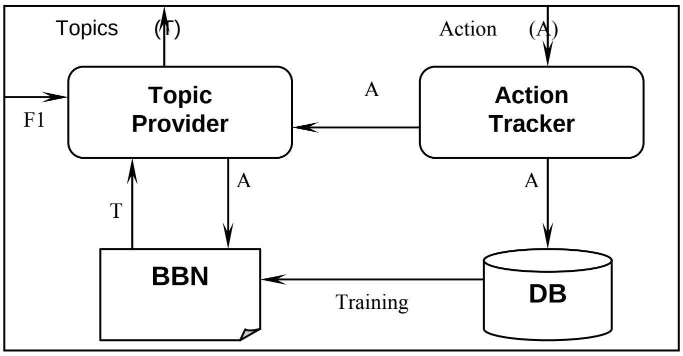

Figure 4. Data exchange between modules in the frame of the Enhanced User Support System of Models Creator.

At run time the developed module operates as follows: Every action of the user is sent to the AUSM module through an appropriate method call. When the user asks for help the Models Creator will call another method of AUSM, that will return the most probable topics for the current state of the interaction. These actions are obtained by instantiating in the Bayesian network the current action node and calculating the probabilities for the next action instances. Each next action is related to a relevant help topic. When a new action is performed the AUSM module writes in the database the data relating to the previous, current and next actions. Afterwards, it sends further the action performed so that the module is able to instantiate the current action node with the current action performed providing this way the adaptive version of user support.

An example of the actual behaviour of the AUSM module is showed in Figure 5. In order to connect two entities with each other, the student has to enter those entities into the main space, select at least an attribute in each entity, drag a relationship and finally connect the edges of the relationship with the desired attributes, by pointing *exactly* at the area of the entity where the verbal representation of the attribute is presented. This procedure posed significant problems to the students, and patterns of interaction showing repeated attempts to properly connect a relationship were detected in the log files obtained. When the user asks for help, while she tries unsuccessfully to connect two objects with a relation, MC's extended version presents the help section proposing three help topics suggested as relevant to the current interaction state. All help topics are presented in a task driven form to help user to carry out specific tasks. That is, the help system is not constrained to a description of each element of the user interface which is of limited help to the user, but it describes in detail how a representative task is carried out using MC's environment. In the specific example described, the three proposed help items explain how to connect a relation (which is actually the desired help item), how to run a model and how to insert a relation into the MC's environment (Figure 5). This approach facilitates considerably the task of finding the desired help item: Instead of selecting from a plethora of available help topics concerning a variety of actions, the user is provided with the most probable three. If the desired help item is not successfully suggested by the AUSM as in this example, the system alternatively permits viewing the help topics, sorted in alphabetical order and grouped by subject.

Module evaluation

Evaluation of the developed system was carried out in three different ways. First, the predictive performance has been measured using the training data set. Secondly, through inspection of the efficiency of the realised AUSM system against log files obtained by another study, and finally with a user observation involving actual users while executing representative tasks with the system.

It is common practice to evaluate BBN predictive performance by testing the network against data that were not used for training it. This is an advantage of such data-driven approach against efficiency-centric and expert-centric approaches for development of an adaptive system (Heckerman, 1996). The effectiveness of the Bayesian Network obtained was validated by using the 10-fold cross validation method. With this method, the data population is randomly divided to training and validation data sets in a 9:1 ratio, repeated ten times. The mean performance of the developed network was found to be 88,43% (+-1,36%, $p<0.05$) which is considered remarkably high considering the relatively limited amount of data used to train the network.

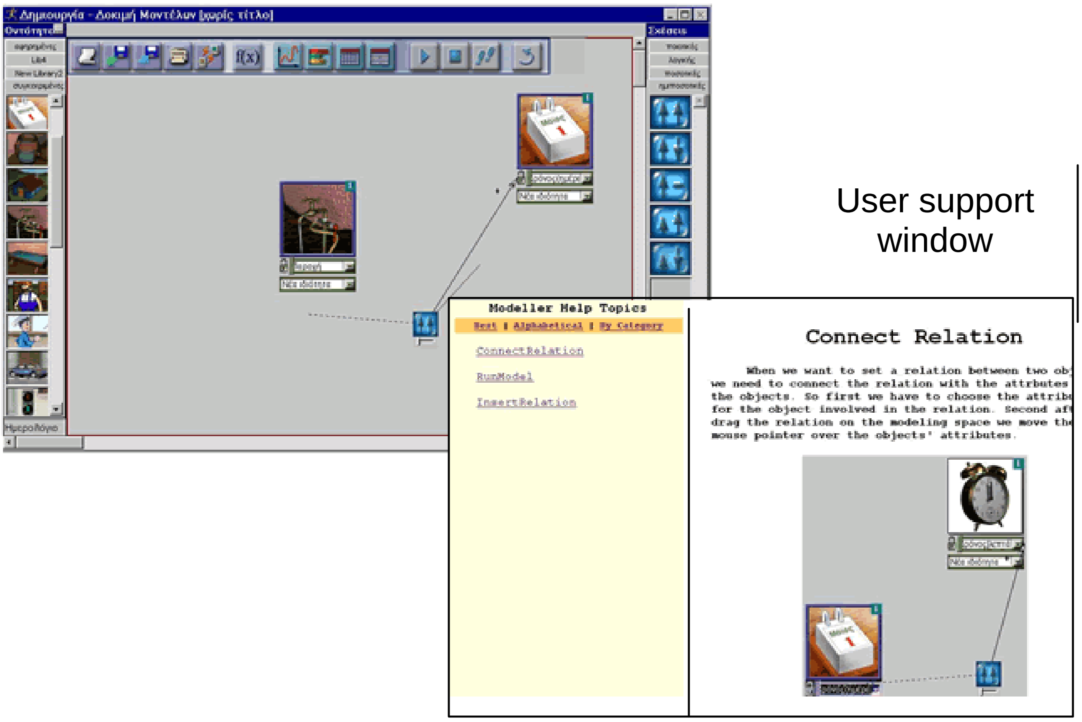

Figure 5. The user interface of the adaptive user support environment. Three options are provided to the user in the help window, with the most probable one actually shown to the user.

As described in the previous section, the Bayesian network structure and conditional probabilities were derived from data collected from 15 log files produced by actual use of the Models Creator software. A new log file was obtained during an experiment in which the activity given to the student to complete was of different nature than that of the tasks in the training data set. This log file contained 141 records of user actions and was used to evaluate and test the adaptive module.

An agent has been implemented to simulate performing the actions contained in the log file and logged the action and the help topics provided by the AUSM module. For every action performed, the module provides the most probable three next actions. The help topics presented by the AUSM are directly related with the predictions in the form: <most probable next action, second most probable action and third most probable action>. Therefore, the quality of the help item presented to the user is directly related to the successful prediction of the next action. The adaptive module inferred successfully the next action and designated as the most probable action in 44,681% of the cases, as the second most probable in 24,113% of the cases and as the third most probable action in 8,511% of the cases. So the overall result is that the adaptive module guessed correctly the next action in 77,305% of the cases.

The results are of the same order with those produced by the mathematical evaluation of the BBN derived, especially if we consider the fact that the task presented here was completely different from the tasks that actually the BBN was constructed from. Extensive testing of several configurations and reformulations for the data used in order to train the Bayesian network, showed that the best results were obtained when we kept track of previous and current action and we tried to guess the next action. In addition to this, taking into account a third action back, led to a very small improvement to the obtained results.

Finally, a user study was contacted, involving 5 students (four male, 1 female), in order to further examine the robustness of the Enhanced User Support Model (AUSM). Representative activities were given to the users to carry out, such as creating a simple model, execution of the model and creation of alternative representations for selected variables of interest of the model. During observation, the users were asked to externalise their thoughts and retrieve help via the AUS Module whenever they had problems carrying out the task. In addition to this, they were asked to evaluate the relevance of the proposed help items in a scale from 1 (not relevant) to 5 (very relevant). 32 calls of help were recorded in total (6,4 per user). The relevance of the proposed help items was 3,358 in average (standard deviation 0,212). After the end of the session, a questionnaire was given to the student, to evaluate effectiveness of the AUSM. The questions and the results of the questionnaire are summarised in Table 1.

No	Question	Mean Value
1	The AUSM helped you carry out specific tasks?	4,33
2	Did you find suitable the approach to present the three more relevant help items?	4
3	How frequently did you find useful the AUSM?	3
4	Did you find the help information presented understandable and in suitable structure?	4
5	To what extend did you find the AUSM useful to you?	3,33
6	The presentation of the three most relevant help items created confusion to you?	1,66 (4,33)
7	Is it difficult to find the desired information item using AUSM?	2,66 (3,33)
Overall Evaluation		3,76

Table 1. Synopsis of the questionnaire referring to the AUSM, and the results obtained (Scale 1-5).

In general, the users evaluated very positively the existence of the Adaptive Help Module, particularly in the case where many alternative tools were available, leading to confusion and long search for help. The discussion with the users, revealed their preference for active exploration of the facilities of the software when they use it for the first time. This statement and the observation that the users avoid reading handbooks prior to direct experimentation with a new software environment has been recorded also by other researchers (Hellman, 1989).

Some specific observations of the user study are discussed next. In some cases, e.g. (a) activity of linking two objects in a model with a relation and (b) the selection of variables to create a graph, the provided help was found to be critical for successful and uninterrupted task execution. This was discovered during the user study. These problems were attributed to fundamental gap between the designer's conceptual model (actual system design) and the user's conceptual model (expectations of the users referring to the actual way to carry out specific actions). In another occasion, the users experienced frustration when they repeatedly tried to change the direction of a specific relation between two objects, an action that is not directly supported by the MC. Finally, the users found the cut and paste tools and functions difficult to use. Concerning the AUSM's presentation of information, they found that the explanations are clear and well written, but they would like to have more links to relative topics in the body of the help text.

Finally, the actions carried out during the usability evaluation of the AUSM were recorded, collected and analyzed. They were analyzed against the relevance of the three help items presented to the user, concerning the actual user's next action. The results were in line with the preliminary evaluation of AUSM. The AUSM presented first the desired help item in the 40,181% of analysed actions, as second in the 27,765% of actions and as third most likely in the 5,869% of cases, with a total forecast success 73,815%. The analysis of the events seems to be in line with the user's subjective evaluation of the AUSM.

Using Bayesian Networks for automatic classification of problem solving strategies

In this section, development of a tool for automatic classification of problem solving strategies is described. The source data for BBN construction were captured from user observation sessions of 30 high school students while using the C.AR.ME microworld. This environment is an open interactive system using multiple representations for learning concepts of geometry, i.e. equivalent shapes and surface measure methods, (Kordaki and Potari, 1998). The experiment concerned activities for solving two distinct problems: (a) transformation of a non-convex polygon into a geometrically equivalent shape and (b) surface comparison of a non convex polygon with a square. Students were asked to solve these two problems with all the possible approaches that might find appropriate, using the tools offered by the microworld. The analysis of the problem solving strategies expressed by the students is extensively described in the work of Kordaki and Potari, (1998). The different problem solution strategies that the students used were classified in thirteen (13) categories. The most common strategy was described as "Using the measurement function", that was applied by 23,7% of the students, while the "Using the automatic measurement function" was applied by 18,4% . These two strategies involved direct use of provided tools. Other strategies involved direct manipulation of shapes and combination of tools with direct manipulation and transformation of the drawn shapes.

114 log files have recorded the sequences of operations of the 30 students to solve the given problems. Each one of them was related to a specific problem solving strategy. These log files were used to construct the BBN. In most of these log files, extensive comments were associated, describing the strategy expressed by the student, accompanied by a screenshot showing the final state of the microworld, at the end of the problem solving process. These log files were analysed using the Usability Analyzer (Tselios et al., 2002), a tool supporting analysis of log files captured from educational environments. In this tool, the evaluator can review the interaction process, by observing simultaneously the captured log files, comments taken during the field study and recorded screenshots showing the student's interaction with the educational software.

The next phase was to store the log files' information into a table. The fields used were the 20 different low level student operations recorded and the frequency of usage for each one in every problem solving strategy. For example, *Cut* and *Paste* operations depict editing actions, *MeasuringAreas* depicts usage of a tool to measure the area of a specific area, and *Triangle*, or *Rectangle* operations depicts usage of automated tools to transform a geometric shape to another type of equivalent area. The hypothesis used to build the BBN and transform accordingly the interaction data, was the hypothesis that whenever a student adopts a specific strategy to solve a problem, he carries out a set of specific operations with high frequency. This kind of activity could be traced in her log file, and could help us classify automatically the problem solving strategy. Additionally information stored, referred to the total number of interaction events (sum variable) and the type of strategy used as recognized originally by the evaluator (class variable, see Figure 6).

BBN Construction

The diagrammatic representation of the BBN obtained is shown in Figure 6. In general, a BBN is able to derive the probability distribution for any given variable subset, for specific values of the remaining variables. From the structure of the network it is shown that the *startdraw* and *enddraw* variables -which are the events created when the student begins or completes a shape's drawing accordingly- are depending the one from the other, but are not connected with the main body of the network. This is not something unexpected, since these events indicate only the start and the end of the user's activity, but not the exact nature of the specific problem solving strategy. In addition, four (4) more types of operations used in the log files, *UnitIteration, Erase, Clear* and *Symmetry* are found to be independent from any other variable and they were excluded from the graph.

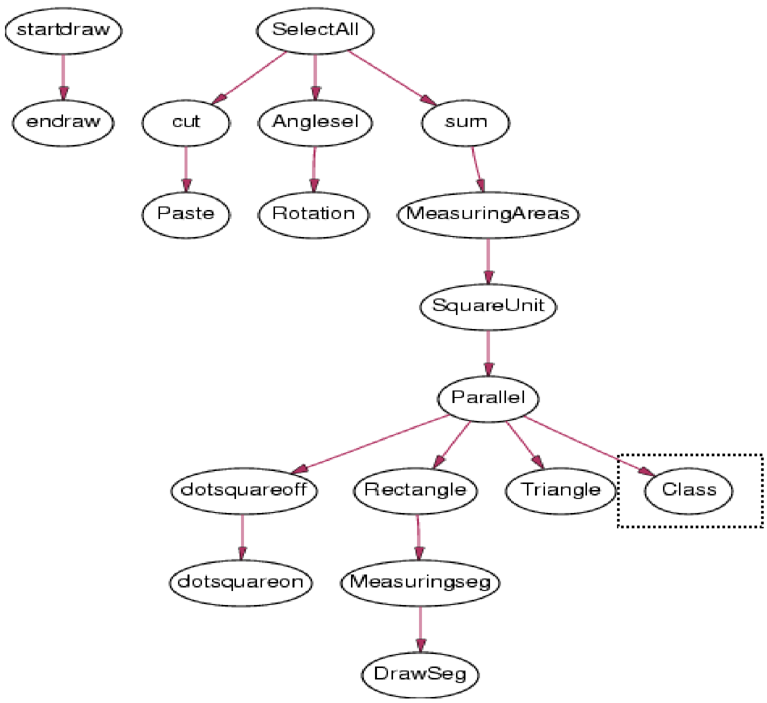

Figure 6. Bayesian network obtained to infer for the problem solving strategy expressed by the student from the log files.

Useful information can be extracted by observing the BBN graph. An example of the use of the network obtained is presented in Figure 7, for specific relative high usage of *triangle* and *square* transformations (Triangle and Square variables have been set to specific values, see Figure 7) the probability distribution in the 'class' variable heavily suggests that the most probable problem solving strategy followed by the student is the C7, i.e. the one that involves usage of automatic measurement function (60% to 9,3% for the second most probable strategy, C5).

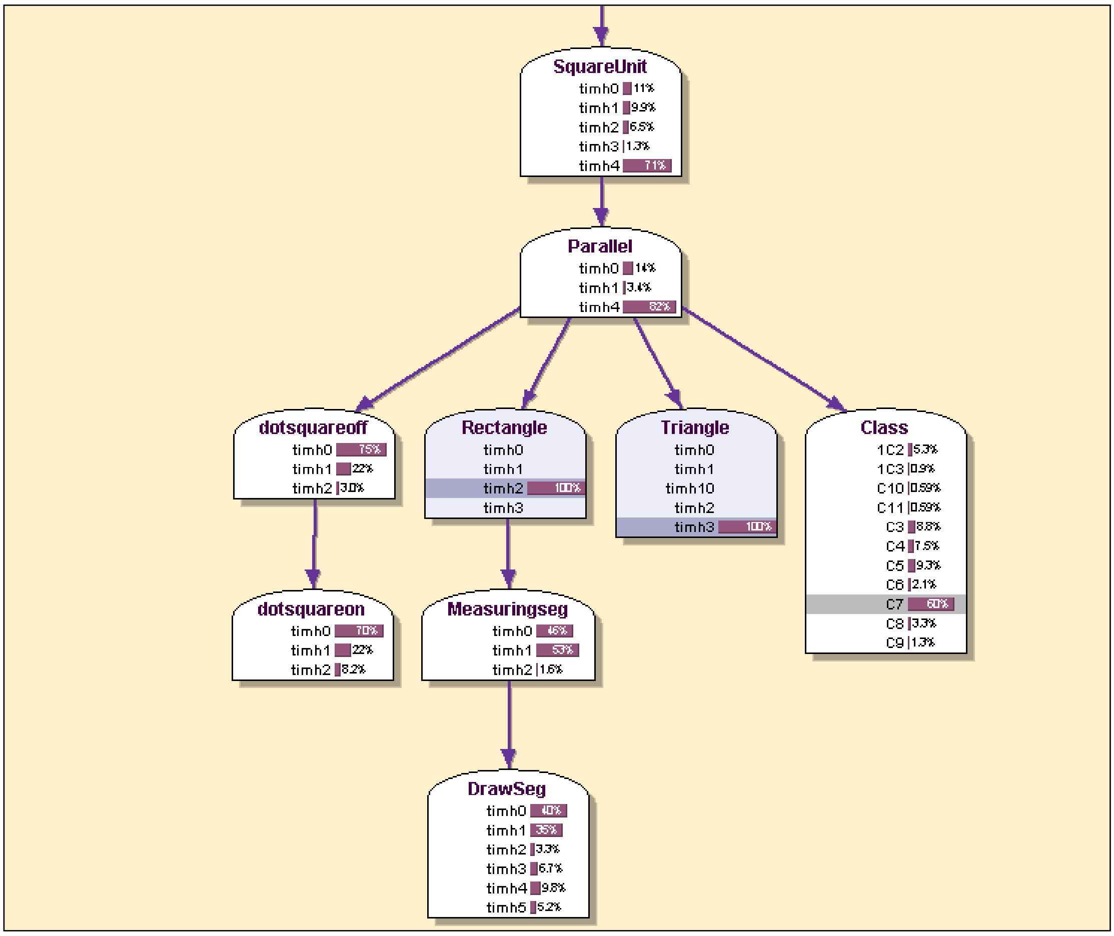

Figure 7. An example of use of the obtained BBN. For given values of variables 'Rectangle' and 'Triangle' the probability the used followed the strategy C7 is very high (above right).

Comparison with other methods

In order to validate the effectiveness of the network, that is the ability to correctly predict the problem solving strategy originally expressed by the student, the ten fold cross validation method has been used. The performance of the BBN was 84,07% (+- 6,75% for $p<0.05$) of correct estimations.

Subsequently, the experiment has been repeated using other popular machine learning techniques such as Decision Tables, C4.5 and Naïve Bayes (Witten and Frank, 2005), which have been included in the automated learning tool constructed by Holmes et al. (1994). The goal was to compare the effectiveness of the BBN method in comparison with other widely accepted techniques. The results are summarized in Table 2. As shown in Table 2 the proposed method using BBN seems to perform better with respect to the other techniques. The only disadvantage compared to Naïve Bayes method is that the latter is easier to implement and apply.

Method	Bayesian networks	Naive Bayes	Decision Tables	C4.5
% success rate	84,07%	72,56%	63,71%	64,60%

Table 2. Performance comparison of categorization techniques

As one can deduct from the above table, Bayesian networks outperform all other methodologies, both of similar nature (such as the statistical-based Naïve Bayes which does not take into account any dependencies between the variables) and the decision-tree based (such as C4.5 and Decision Tables). Regarding the former case, Bayesian networks alleviate the over-restrictive assumption on the conditional independency on the attributes of the naïve Bayesian classifier, which is also often unrealistic. Bayesian networks, can reason under conditions of uncertainty, taking the semantic interrelation of attributes into account. Concerning the decision-tree based algorithms, we could argue that information gain, a metric that is the core of the algorithms can provide a clear view on the value of each attribute but can also lead to large trees, in which the task of pruning some redundant nodes is difficult and may cause the stall of the algorithm in local maxima. On the other hand, Bayesian networks are based on the probability of a structure over the given training set, which may be an NP-hard problem but using intelligent search strategies can overcome the problem of local maxima.

Conclusions

This paper presents the application of Bayesian Networks techniques for tackling two important problems in the frame of two open problem solving environments (MC and CARME), with very promising results. In the first case, we discussed the design and implementation of the adaptive user support module AUSM. The performance of the developed module during the evaluation experiment was very promising. The system, when fed with interaction data of a new problem solving task was able to predict correctly 3 out of 4 cases the next user action and therefore to provide the user potentially with useful support in an efficient way. The proposed approach serves as a scaffolding mechanism during modelling activities. The support system helps students accomplish a complex task requiring specific actions, at critical points of their task. The proposed scaffolding mechanism, realised with the use of Bayesian Networks, does not reduce the transparency of the system. Instead, it provides the ability to the student to move ahead over potential critical points, in terms of tools operations. In the scaffolding process adopted, the system provides help to the students on those elements of tasks that are beyond their capacity, and allow them to concentrate upon the elements that are within their range of competence. This is accomplished with identification of frequent interaction patterns using Bayesian Networks and subsequent correlation with desired task accomplishment, thus enabling the system to infer the desired help item. It provides help just when it is needed thus reducing the effort to search for relevant support, thus maintaining focus on the modelling process without interruption and disorientation.

In the second case of automatic classification of problem solving strategies in an open problem solving environment, the approach produced good results. The classification rate of the BBN algorithm is very high in comparison with other popular algorithms, even with a small amount of data provided. However, the BBN could be proven unsuccessful to recognize strategies with low occurrence rate represented in the reference sample. This is a future research goal, to estimate the lowest possible sample for each category related to the total, which lead to unbiased and solid results. Various techniques have been proposed in the literature for tackling this problem (e.g. Daskalaki et al. 2006), these could be particularly useful in our case, since often rare problem solving strategies, are of particular interest and their diagnosis is required by the tutors.

Another important advantage of the Bayesian Network approach is the derived graph which offers an easy to understand representation of the variables showing the nature of the interaction. Study of the graph could lead to usability improvements, as shown in the first study, by providing the most suitable help items or the possible actions related to the interaction dialogue flow and status. From an educational perspective, automated problem solving strategy recognition in open problem solving environments could substantially improve the learning process and support the tutors, when dealing with large sets of student data. That is, the educator could rapidly verify the adoption of specific problem solving strategies. Additionally to this, real time implementation of the proposed approach, in the case of distance learning systems could aid for diagnosing and evaluating of the learning outcome and support teacher intervention in the form of adequate feedback.

In general, it is argued that a more efficient and adaptive user system increases indirectly the pedagogical value of the environment because it contributes to the transparency of the tool, a more intuitive flow of interaction and enhances the learnability of complex tools such as those included in open problem solving environments. Our approach could be expanded to adapt the behaviour of the open educational environment in various aspects such as goal recognition, adaptive assessment regarding the expertise of the user, and adaptation of the interface (for example, adaptation of right click pull down menus, together with a constant set of commands). Further research is needed in order to investigate these areas, together with longitudinal evaluation of the effects of the developed module on the way it affects student learning within such an environment.

References

Avouris, N., Tselios, N. & Tatakis E.C. (2000). Development and Evaluation of a Computer-based Laboratory Teaching Tool, Journal Computer Applications in Engineering Education, 2000 (accepted for publication).

Bunt, A. & Conati, C. (2003). Probabilistic Student Modelling to Improve Exploratory Behaviour. Journal of User Modeling and User-Adapted Interaction, vol 13 (3), p. 269-309.

Collins, J., Greer, J. & Huang, S. (1996). Adaptive Assessment Using Granularity Hierarchies and Bayesian Nets. In Proceedings of Intelligent Tutoring Systems, pages 569-577,1996.

Conati, C., Gertner, A.S., VanLehn, K. & Druzdel M. (1997). On-line Student Modeling for Coached Problem Solving Using Bayesian Networks. In Proceedings of the Seventh International Conference on User Modeling, pp 231-242, 1997.

Conati, C., Gertner, A. & VanLehn, K. (2002). Using Bayesian Networks to Manage Uncertainty in Student Modeling. Journal of User Modeling and User-Adapted Interaction, vol. 12(4), p. 371-417.

Cooper, J. & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. Machine Learning, 9, pp.309-347.

Daskalaki, S., Kopanas, I. & Avouris N., (2006). Evaluation of Classifiers for an Uneven Class Distribution Problem, Applied Artificial Intelligence, 20:1–37.

Dimitracopoulou, A. & Komis, V. (2005). Design principles for the support of modelling and collaboration in a technology-based learning environment. International Journal of Continuing Engineering Education and Lifelong Learning 2005 - Vol. 15, No.1/2 pp. 30 - 55.

Dimitracopoulou, A., Komis, V., Apostolopoulos, P. & Politis, P., (1999). Design principles of a new modelling environment for young students, AI-ED Proc, 9th Int. Conf. on AI in Education, Le Mans, France.

Ergazaki, M., Komis, V. & Zogza, V. (2005). High-school Students' Reasoning while Constructing Plant Growth Models in a Computer-Supported Educational Environment, International Journal of Science Education, Vol. 27, No 2, June 2005, pp. 909-933.

Glymour, C. & Cooper, G. (eds.), (1999). Computation, Causation & Discovery. AAAI Press/The MIT Press.

Gudzial, M. J. (1993). Deriving Software Usage Patterns from Log Files Georgia Institute of Technology. GVU Center Techical Report.Report #93-41,1993.

Heckerman, D. (1996). A Tutorial on Learning Bayesian Networks. Technical Report MSR-TR-95-06, Microsoft Research, November, 1996.

Hellman, R. (1989). User Support: Revealing Structure Instead of Surface, Behaviour and Information Technology, v.8, n.6, p.417-435.

Holmes, G., Donkin, A. & Witten, I.H. (1994). Weka: a machine learning workbench. Proceedings of the 2nd Australian and N. Zealand Conference on Intelligent Information Systems, Brisbane, Australia, pp. 357-361.

Jameson, A. (1995). Numerical uncertainty management in User and Student Modeling : An Overview of Systems and Issues. In User Modeling and User-Adapted Interaction 5, 1995.

Jeffreys, H. (1939). Theory of Probability. Clarendon Press, Oxford.

Kinshuk, P.A. & Russell, D. (2002). Intelligent and Adaptive Systems. In Adelsberger, H., Collins, B., and Pawlowski, M.J. (Eds.) Handbook on Information Technologies for Education and Training, Germany: Springler-Verlag, 79-92.

Komis, V., Dimitracopoulou, A., Politis, P. & Avouris, N., (2001). Expérimentations exploratoires sur l'utilisation d'un environnement informatique de modélisation par petits groupes d'élèves, Sciences et Techniques Educatives, Vol. 8, no 1-2, April 2001, pp.75-86.

Komis, V., Avouris, N. & Fidas, C. (2002). Computer Supported collaborative concept mapping: Study of Interaction, Education and Information Technologies, 7:2, pp. 169-188.

Kordaki, M., & Potari, D. (1998). A learning environment for the conservation of area and its measurement: a computer microworld. Computers and Education, 31, 405-422.

Laborde, J-M. (1990). Cabri-Geometry [Software]. France: Universite de Grenoble.

Luger, G.F. & Stubblefield, W.A. (1998). Artificial Inteligence – structures and strategies for complex problem solving, Third Edition, Addison Wesley, 1998.

Mayo M. & Mitrovic A (2000). Optimizing ITS Behaviour with Bayesian Networks and Decision Theory. In International Journal of Artificial Intelligence in Education , 2000 (12).

Nathan, M.J. (1998). Knowledge and Situational Feedback in a Learning Environment for Algebra Sotry Problem Solving. Interactive Learning Environments 5(1). pp. 135-159.

Niedermayer, D. (1998). An Introduction To Bayesian Networks And Their Contemporary Applications, University of Saskatchewan, Technical Report 184-3-54440 1998.

Norman, D.A. (1986). Cognitive Engineering in User-Centred System Design, Laurence Erlbaum Associates, Hillsdale NJ, 1986.

Papert, S. (1980). Mindstorms: Pupils, Computers, and Powerful Ideas. New York: Basic Books.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA: M.K.

Squires, D. & Preece, J. (1999). Predicting Quality in Educational Software: Evaluating for Learning, Usability and the Synergy between them, Interacting with Computers, 11, 1999 , pp. 467-483.EEE Proceedings, F. T. Ulaby, S.W.Director (guest editors), Special Issue on Electrical and Computer Engineering Education, vol 88 (1), January 2000.

Stephenson, T.A. (2000). An Introduction To Bayesian Network Theory And Usage , Institut Dalle Molle d'Intelligence Artificielle Perceptive, Technical Report IDIAP-RR 00-03, 2000.

Tselios N., Avouris N. & Kordaki M. (2002). Student Task Modeling in design and evaluation of open problem-solving environments, Journal of Education and Information Technologies, 7(1), pp. 19-42.

Suthers, D., Connely J., Lesgold, A., Paolucci, M., Toth, E.E., Toth, J. & Weiner, A. (2001). Representational and Advisory Guidance for Students Learning Scientific Inquiry. In Forbus, K. and Feltovich, P. (EDs.), Smart Machines in Education. The MIT Press. 2001, ISBN 0-262-56141-7

Von Glasersfeld, E. (1987). E. Learning as a constructive activity. In Janvier, C. (Ed.) Problems of representation in teaching and learning of mathematics London: Lawrence Erlbaum associates, 1987, pp. 3-18.

Vomlel, J. (2004). Bayesian networks in educational testing. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 12, pp. 83-100.

Xenos, M. (2004). Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks. Computers & Education, 43(4), pp. 345-359.

Witten,I. & Frank, E.(2005). Data Mining: Practical Machine Learning Tools and Techniques, Second Edition. Morgan Kaufmann Publishers, San Francisco, USA.

Woolf, B.P., Beck J., Eliot, C. & Stern, M. (2001). Growth and Maturity of Intelligent Tutoring Systems. In Forbus, K. and Feltovich, P. (EDs.), Smart Machines in Education. The MIT Press. 2001, ISBN 0-262-56141-7.