UTILIZING DIAGNOSING PROBLEMS IN A PROBABILISTIC DOMAIN TO BUILD STUDENT MODELS

Nabila Khodeir[1], Nayer Wanas[1,2], Nevin Darwish[3],
Nadia Hegazy[1]

[1]Informatics Dept., Electronics Research Institute, Tahrir St., Giza, Egypt
nabilakhodeir@yahoo.com, nhegazy@mcit.gov.eg
[2] Cairo Microsoft Innovation Lab, 306 Chorniche El-Nile, Maadi, Cairo, Egypt
nayerw@microsoft.com
[3]Dept. of Computer Engineering, Faculty of Engineering, Cairo University, Giza, Egypt
ndarwish@ieee.org

ABSTRACT

In this paper we aim to estimate the differential student knowledge model in a probabilistic domain within an intelligent tutoring system. The student answers to questions requiring diagnosing skills are used to estimate the actual student model. Updating and verification of the model are conducted based on the matching between the student's and model answers. Two different approaches to updating are suggested, i) coarse and ii) refined model updating. Moreover, the effect of the order of which questions are presented to the student is investigated. Results suggest that the refined model, although takes more computational resources, provides a slightly better approximation of the student model. In addition, the accuracy of the algorithm is highly insensitive to the order of which the questions are presented, more so when using the refined model updating approach..

KEYWORDS

Bayesian networks, Abduction, Intelligent tutoring system, Student modelling.

1. INTRODUCTION

The student model is a core component in any intelligent or adaptive tutoring systems. It allows individual personalization of the system resources and intervention for each student. The student model can represent one or more of the following student features i) knowledge[1], ii) interests[2,3], iii) goals[4, 5], and iv) individual traits[6, 7, 8]. The student's knowledge stands out as the most important feature in tutoring system that aim to evolve and assess the student knowledge in a specific domain.

Several models have been suggested for the student knowledge modeling namely i) the overlay model, ii) the perturbation model and iii) the differential model. The overlay model represents the user model as a subset of the expert model [9,10]. The perturbation model extends the overlay model by adding representation of the student incorrect knowledge (misconceptions or bugs). This is usually achieved by adding predefined library of the students' bugs [11, 12, 13, 14]. The differential model, on the other hand, represents both the student knowledge and the differences between student and expert knowledge [15, 16]. That is to say, the differential model is basically an overlay on expected knowledge, what's more it is instance of the perturbation model because expected knowledge can be considered as the knowledge that user lacks. We will invoke the idea

DOI : 10.5121/ijcsit.2010.2408

of differential model in our proposal, but we will modify it to include not only the lack of knowledge but also the erroneously added knowledge.

The majority of adaptive tutoring systems that model the student knowledge deal with deterministic domains (domains that have deterministic rules and relations between their items). However, many practical domains have some degree of uncertainty. Among these domains is the medical domain which has uncertainty or probability in the relations between its items and consequently solutions of its problems. In this paper we will focus on exploiting the structure of such domains to infer the differential student model. The structure of the domain greatly influences on modeling of the student knowledge. For example, the overlay model for domains that decomposed into a set of elements is a set of masteries over these elements. If these domain elements imply interrelations between each other, such as prerequisite relations, that allows inferring more knowledge about the student through these relations [17, 18, 19]. With respect to the probabilistic domains, Bayesian network is a widely used approach to represent such domains to handle their uncertainty of their relations. Suebnukam and Haddawy [20] suggested an algorithm that focuses on the skill of reasoning around practical patient problems. In this work, we deal with the diagnosing skill to infer the student knowledge, especially for ambiguous cases where there are more hypothesises that explain the symptoms presented by the patient.

Bayesian Networks (BNs) are probabilistic graphical models that can be considered as a form of knowledge representation. BNs allows for the definition of the causal relations between different variables using links. Moreover, it allows the strength of these relations to be defined using the Conditional Probability Distribution (CPD) at each node. The CPD lists the probability that the child node takes on each of its different values for each combination of values of its parents. In this work, the medical domain is used as an example of probabilistic domains. The relations between diseases and symptoms have causal relations with probabilities in addition to the prevalence of each disease are represented as a Bayesian network. This representation is exploited to update the student model. Moreover it is utilized to verify the updating of the student model.

Six diseases from paediatric medical domain are selected based on their community of some symptoms. Structure of the Bayesian network and their conditional probability are obtained from domain experts.

Our contribution is a proposed method to build a differential student model in a probabilistic domain based. A Bayesian network is used to represent both the domain and the student model. Based on the domain structure we generate problems that require diagnostic skills to be solved. The discrepancies between the answers generated by the student model and the answers provided by the student are used to update the student model. It is worth noting that the domain knowledge and student models are represented using BNs. This allows for simple retrieval of information about the student correct and incorrect knowledge, without the needs to predefine a library of bugs.

The reminder of this paper is organized as follows. Section 2 presents the proposed approach for building and updating student models in a probabilistic domain. Thereafter we explore the experimental results that show the performance evaluation of the algorithm implementation in section 3, discussion of the result and conclusion in section 4.

2. Estimating Student knowledge Model in Probabilistic Domains

Starting with an initial student model, a question that describes a case with some symptoms is presented to the student. The student is expected to choose and rank the most likely diagnosis diseases from a list. The question is generated using the domain knowledge such that two or three symptoms in the questions that are common between at least two diseases. It is worth mentioning that this question form is selected since the correct answer is tightly coupled, not only to the knowledge structure (which means the relations between diseases and symptoms), but also on the weights of the relations between the knowledge items. Moreover, the answer of this question can be obtained automatically by applying abduction inference through the Bayesian network. In this work we use the abduction algorithm suggested by Nilsson [21, 22]. Subsequently, the answer provided by the student to a specific question is compared to that generated by the student model using the abduction algorithm. The algorithm is invoked on the student model represented as a Bayesian network with the symptoms stated in the question acting as evidence. The output of the algorithm is managed to give a ranked list of hypotheses. If the answers provided by the student and the algorithm match, then the student model doesn't require regulation. On the other hand, if there is a discrepancy between the answers the student model is regulated.

Since the answer is a ranked list of hypotheses, the difference between the two answers can be one of the following i) missing one or more hypothesis, ii) adding one or more hypothesis, iii) answers are presented in a different order, and iv) a combined error of missing and added hypotheses in a correct or incorrect order. Identification of the type of difference can be associated with the difference between the student model and student knowledge. The addition or absence of a hypothesis is associated to a difference in the relations between domain items. On the other hand, the difference between the hypotheses order is associated to the difference in the weight of theses relations. Upon identifying the type of difference, the student model should be updated accordingly.

Two approaches are suggested to update the student model, namely (i) coarse model update and (ii) refined model update. Model verification is evaluated to verify that the new model matches the answer provided by the student. It is worth mentioning that matching between the updated model and the answer is used to terminate the algorithm. Matching is performed at two levels (i) unordered matching of hypotheses and (ii) ordered matching.

If the first level match is successful, the structure of the student model is assumed to be correct. The second level match indicates that the weights of the relationships and prevalence are adjusted correctly. Repetition of regulation of the student model is gradually continued until the second level match is successful. Regulation will be terminated if the difference between the answers increases or the weights reach the threshold values for all regulated relations. In case the model update fails to achieve a match between the two answers, the student model is left unchanged.

2.1. Coarse Update

The coarse update is conducted by either i) adding or removing of relations responsible for the differences in the hypotheses and ii) swapping between relations' weights and/or prevalence for the differences in the hypotheses order. These updates are based on the type of error detected as follows:

- *Missing disease(s):* the student model is regulated by establishing missing relations with the weight assigned to be equal 0.5.
- *Adding disease(s):* the student model is regulated by removing the existing relations between the hypotheses added and the mentioned symptoms.
- *Answers presented in an erroneous order:* the student model is regulated by swapping the weights of the relations and prevalence between the two diseases that are conflicted with each other.
- *Combined error:* the regulation proceeds first on the missing and adding hypotheses. After level one match is achieved, the regulation proceeds on the order difference.

2.2. Refined Update

The refined update is performed using successive increase or decrease in the weights and prevalence of the different hypotheses according to the following

- *Missing disease(s):* the student model is regulated by increasing the relations' weights between the missed hypotheses and the symptoms that are mentioned in the question. If one or more of these relations are absent, the relations are established by initial value equal to 0.01. Theses relations' weights are increased by small increments until the models match or the weight takes the maximum value.
- *Adding disease(s):* the student model is regulated by decreasing the weight of the relations between the added hypotheses and the symptoms under investigation. Theses relations decreased by small decrements until the models match or the weight take the minimum value.
- *Answers presented in an erroneous order:* the student model is regulated by increasing the relations' weights between the disease with the lowest rank and the symptoms. In addition, the weights of the relations between the disease with the highest rank and the symptoms are decreased. If that failed to achieve matching the student model is updated by increasing the prevalence of the disease that takes lowest rank and decreasing the prevalence of the disease with the highest rank.
- *Combined error:* the regulation proceeds first on missing and adding hypotheses and after level one match, then regulation proceeds on the order.

It is worth mentioning that the difference between the modified student model with the domain model represents the student incorrect knowledge.

3. Evaluation of the Student Model Estimation

We used simulated students for the evaluation of the algorithm. Collins et al.[23], Van Lehn et al.[24], and Millan et al.[25] have suggested similar simulation based approaches for evaluating student models. Simulated students give us the capability of measuring the difference between the simulated student and the obtained differential user modeled after applying the proposed algorithm. Simulated students are automatically generated and modelled by different Bayesian networks that represent the students' knowledge. Generation of simulated students is based on two different schemes for perturbation of the knowledge model, namely (i) fully random and (ii) partially random schemes. The fully random scheme performs perturbation of the knowledge model through changing the relations (adding and removing relations) and relation weights randomly for all domain items. On the other hand, in the partially random scheme only the weights of the relations for two randomly chosen domain items are altered (The concepts related to the questions being tested). The target of the algorithm is to adjust the initial student model stored to be similar to the generated model representing the actual student.

To test the performance of the algorithm five groups of twenty different questions are automatically generated. The questions are evaluated against the initial student model, generating the original match. The performance of the algorithm is evaluated after each question (individual match). Moreover, the set of twenty questions are all tested against the new adjusted model for verification (verification match).

Tables 1 to 3 outline the performance for the algorithm with different student and initial models. It is observed that the algorithm is capable of correcting the student model to achieve at least 84% and 68% accuracy on the individual and verification matches respectively, even when the student model and model for the actual student are selected at random. Limiting the degree of difference between the student and the student model allows the accuracy to improve to almost 98% and 88% for the individual and verification matches respectively. It is worth mentioning that the coarse and refined model updating have a similar performance with respect to accuracy. However, using the refined model update produces a BN that is closer to the target model compared to the coarse model update. This is indicated with the 8% reduction in the summation of the absolute difference between the updated model and target model where we assign zero for the missed relation's weight.

Table 1. The performance of the algorithm with the initial student model set to the knowledge model and a fully random model representing the actual student.

	Original match	**Coarse Model Update**		**Refined Model Update**	
		Individual	Verification	Individual	Verification
Total match	0	19.6 ± 0.44	16.0 ± 1.34	19.6 ± 0.36	15.8 ± 1.33
Total mismatch	20	0.4 ± 0.44	4.0 ± 1.34	0.4 ± 0.36	4.2 ± 1.33
Accuracy%	0	98.0 ± 2.18	79.8 ± 6.72	97.8 ± 1.79	79.2 ± 6.67

Table 2: The performance of the algorithm with the initial student model set to a random model and a fully random model representing the actual student.

	Original match	**Coarse Model Update**		**Refined Model Update**	
		Individual	Verification	Individual	Verification
Total match	9.2 ± 3.99	16.9 ± 3.42	13.8 ± 2.92	16.8 ± 3.47	13.9 ± 2.74
Total mismatch	10.8 ± 3.99	3.1 ± 3.42	6.2 ± 2.92	3.2 ± 3.47	6.1 ± 2.74
Accuracy%	45.8 ± 19.95	84.4 ± 17.10	68.8 ± 14.62	84.0 ± 17.33	69.4 ± 13.69

Table 3: The performance of the algorithm with the initial student model set to the knowledge model and a partially random model representing the actual student.

	Original match	**Coarse Model Update**		**Refined Model Update**	
		Individual	Verification	Individual	Verification
Total match	11.7 ± 6.36	19.1 ± 1.18	17.2 ± 1.96	19.7 ± 0.52	17.7 ± 1.69
Total mismatch	8.3 ± 6.36	0.9 ± 1.18	2.8 ± 1.96	0.3 ± 0.52	2.3 ± 1.69
Accuracy%	58.6 ± 31.79	95.6 ± 5.90	86.0 ± 9.82	98.4 ± 2.61	88.6 ± 8.47

In order to investigate the effect of the order by which the questions are presented to the users, we have experimented with different question orders. Based on the original match, the questions could be ordered differently prior to being presented to the algorithm, namely

- Original Order: Questions are applied in the same order irrespective of their correctness.
- Order First: Wrongly answered questions are applied first.
- Order Last: Wrongly answered questions are applied last.
- Interleaved Order: Wrongly answered and correctly answered questions are interleaved.
- Order Middle: Wrongly answered questions are inserted in the middle of the list.
- Random Order: Questions are applied in a random order.

Tables 4 to 7 present the effect of presenting the questions in different orders to the model updating algorithm. The results suggest that the effect of different ordering is limited. As outlined in the tables the results of using the refined model updating performs better than the coarse updating. This is also coupled with the fact that the performance using a partially random model is almost 10% better in accuracy compared to using the totally random model. Moreover, when starting from the knowledge model generates performance better than using a totally random model.

Table 4: The (a) individual and (b) verification performance of the algorithm with different ordering of questions using coarse model updating, with the initial student model set to a random model and a totally random models representing the actual students.

	Original	First	Last	Interleave	Middle	Random
Total match	16.9 ± 3.4	16.9 ± 3.5	16.9 ± 3.6	16.8 ± 3.5	17.0 ± 3.3	16.9 ± 3.3
Total mismatch	3.1 ± 3.4	3.1 ± 3.5	3.1 ± 3.6	3.2 ± 3.5	3.0 ± 3.3	3.1 ± 3.3
Accuracy%	84.4 ± 17.1	84.3 ± 17.5	84.3 ± 18.2	84.0 ± 17.9	84.9 ± 16.8	84.7 ± 16.9

a

	Original	First	Last	Interleave	Middle	Random
Total match	13.8 ± 2.9	13.9 ± 2.8	14.0 ± 2.9	13.6 ± 2.9	13.8 ± 2.6	14.2 ± 3.0
Total mismatch	6.2 ± 2.9	6.1 ± 2.8	6.0 ± 2.9	6.4 ± 2.9	6.2 ± 2.6	5.8 ± 3.0
Accuracy%	68.8 ± 14.6	69.3 ± 14.2	69.9 ± 14.6	68.2 ± 14.9	68.9 ± 13.2	70.7 ± 15.1

b

Table 5: The (a) individual and (b) verification performance of the algorithm with different ordering of questions using coarse model updating, with the initial student model set to a knowledge model and a partially random model representing the actual students.

	Original	First	Last	Interleave	Middle	Random
Total match	19.1 ± 1.1	19.2 ± 1.2	19.4 ± 1.0	19.1 ± 1.2	19.1 ± 1.2	19.1 ± 1.0
Total mismatch	0.9 ± 1.1	0.8 ± 1.2	0.6 ± 1.0	0.9 ± 1.2	0.9 ± 1.2	0.9 ± 1.0
Accuracy%	95.6 ± 5.9	95.7 ± 6.2	96.9 ± 4.9	95.3 ± 6.0	95.5 ± 6.1	95.6 ± 4.8

a

	Original	First	Last	Interleave	Middle	Random
Total match	17.2± 1.9	17.0 ± 3.0	17.0 ± 2.3	17.1 ± 3.0	16.7 ± 3.1	16.8 ± 2.6
Total mismatch	2.8 ± 1.9	3.0 ± 3.0	3.0 ± 2.3	2.9 ± 3.0	3.3 ± 3.1	3.2 ± 2.6
Accuracy%	86.0± 9.2	84.9 ± 15.2	85.2 ± 11.7	85.5 ± 14.9	83.5 ± 15.2	84.2 ± 13.0

b

Table 6: The (a) individual and (b) verification performance of the algorithm with different ordering of questions using the refined model updating, with the initial student model set to a random model and a fully random model representing the actual students.

	Original	First	Last	Interleave	Middle	Random
Total match	16.8 ± 3.5	16.9 ± 3.3	16.9 ± 3.5	16.8 ± 3.5	16.8 ± 3.5	16.9 ± 3.3
Total mismatch	3.2 ± 3.5	3.1 ± 3.3	3.1 ± 3.5	3.2 ± 3.5	3.2 ± 3.5	3.1 ± 3.3
Accuracy%	84.0 ± 17.3	84.7 ± 16.6	84.5 ± 17.4	84.0 ± 17.3	84.2 ± 17.3	84.5 ± 16.5

a

	Original	First	Last	Interleave	Middle	Random
Total match	13.9 ± 2.7	14.2 ± 2.3	14.3 ± 2.5	13.8 ± 2.7	14.0 ± 2.4	14.7 ± 2.2
Total mismatch	6.1 ± 2.7	5.8 ± 2.3	5.7 ± 2.5	6.2 ± 2.7	6.0 ± 2.4	5.3 ± 2.2
Accuracy%	69.4 ± 13.7	70.9 ± 11.7	71.3 ± 12.4	69.1 ± 13.4	70.0 ± 11.9	73.4 ± 19.2

b

Table 7: The (a) individual and (b) verification performance of the algorithm with different ordering of questions using the refined model updating, with the initial student model set to a knowledge model and a partially random model representing the actual students.

	Original	**First**	**Last**	**Interleave**	**Middle**	**Random**
Total match	19.7 ± 0.5	19.7 ± 0.6	19.8 ± 0.4	19.6 ± 0.6	19.7 ± 0.6	19.6 ± 0.6
Total mismatch	0.3 ± 0.5	0.3 ± 0.6	0.2 ± 0.4	0.4 ± 0.6	0.3 ± 0.6	0.4 ± 0.6
Accuracy%	98.4 ± 2.6	98.6 ± 2.8	99.1 ± 1.9	98.1 ± 3.0	98.3 ± 3.2	97.9 ± 3.0

a

	Original	**First**	**Last**	**Interleave**	**Middle**	**Random**
Total match	17.7 ± 1.7	17.3 ± 3.0	17.6 ± 1.9	17.4 ± 2.9	17.1 ± 3.0	17.2 ± 2.5
Total mismatch	2.3 ± 1.7	2.7 ± 3.0	2.4 ± 1.9	2.6 ± 2.9	2.9 ± 3.0	2.8 ± 2.5
Accuracy%	88.6 ± 8.5	86.3 ± 15.0	87.9 ± 9.6	87.0 ± 14.4	85.6 ± 15.0	85.9 ± 12.3

b

4. DISCUSSION AND CONCLUSION

In this paper, we presented an algorithm to approximate the student model in a probabilistic domain. The algorithm utilizes diagnosing questions to infer the student knowledge. The algorithm is invoked when the student answer mismatches the expected answer evaluated using the student model. Two modes of update are suggested, namely the (i) coarse and (ii) refined model updating approaches.

An experimental evaluation of the approaches has been conducted. Random models for the student and the student models are generated. A series of twenty questions are also automatically generated and presented to the system based on the domain structure. The performance of the algorithm is evaluated using both the accuracy of estimating the student answers to the questions and the absolute difference between the generated and actual model of the student. The differences between the two models are measured in terms of discrepancies between relations weights. The results suggested that the refined model updating exhibits similar performance with respect to accuracy compared to the coarse updating. However, it estimates models that are closer to the actual model by at least 8% compared to the coarse updating model. It is worth mentioning that the refined updating of the model takes a considerable amount of time compared to the coarse updating model due to its iterative nature. To improve the performance, coarse updating might be applied when the original match is limited, which would reflect poorly performing students. On the other hand, the refined model updating maybe applied to good students, where the search space utilized is limited, or upon doing a few increments with the coarse updating model.

Applying the questions in any order doesn't have a significant effect on the overall performance of the algorithm. On the other hand, the proximity of the initial model selected to that of the student, or to the knowledge model improves the performance of this approach significantly. It is worth mentioning that even when using totally random models for both the students and the student model the approach is successful in estimating the student model with an accuracy of over 84% for the matching of the answer after ever question, and 70% for verifying all the questions upon the last update.

REFERENCES

[1] Brusilovsky, P., Millan, E.:User, (2007) "Models for Adaptive Hypermedia and Adaptive Educational Systems." In: Brusilovsky, P., Kobsa, A., Nejdl, W.(eds.) The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, Springer, Heidelberg, pp. 3-53.

[2] Farzan, R., Brusilovsky, and P. (2005) "Social Navigation Support Through Annotation-Based Group Modeling." UM 2005. Lecture Notes in Computer Science, vol. 3538, Springer, Heidelberg, pp. 463-472.

[3] Smith, A., Blandford, A. (2003) "ML Tutor: An Application of Machine Learning Algorithms for an Adaptive Web-based Information System.", International Journal of Artificial Intelligence in Education., 13, pp. 235–261.

[4] Mhlenbrock, M. , Scheuer, O. (2005) "Using Action Analysis in ActiveMath to Estimate Student Motivation." In: Proceedings of the 13th Annual Workshop of the SIG Adaptively and User Modelling in Interactive Systems, Saarbrcken, Germany, pp. 56–60.

[5] Roll, I., Baker, R., Aleven, V., Koedinger, K. (2004) "What goals do students have when choosing the actions they perform?." In: Sixth International Conference on Cognitive Modeling (ICCM), Pittsburgh, PA: Carnegie Mellon University/University of Pittsburgh, pp. 380–381.

[6] Carmona ,C. ,Castillo , G. , Millan, E. (2008) "Designing a Dynamic Network for Modeling Stu- dents' Learning Style." In: Proceedings of Eighth IEEE International Conference on Advanced Learning Technologies,IEEE Press, p. 346 - 350.

[7] Kelly, D. , Tangeney, B., (2005) "First Aid for Youapos': Getting to know your Learning Style using Machine Learning. In: Fifth IEEE International Conference on Advanced Learning Technologies, 5(8), IEEE Press, Kaohsiung, Taiwan, pp. 1–3.

[8] Stash, N. , De Bra, P., (2004) "Incorporating Cognitive Styles In AHA! (The Adaptive Hypermedia Architecture" Pr IASTED International Conference WEB-BASED EDUCATION,Innsbruck, Austria, pp. 378–383.

[9] Melis, E., Siekmann, J., (2004) "ActiveMath: An Intelligent Tutoring System for Mathematics." In: Rutkowski, L., Siekmann,and J., Tadeusiewicz R., Zadeh, L. (eds.) Lecture Notes in Computer Science, vol. 3070, Springer, Heidelberg, pp. 91-101.

[10] Henze, N., Nejle, and W., (2001) "Adaptation in Open Corpus Hypermedia.", International Journal of Artificial Intelligence in Education., 12, pp. 325–350.

[11] Merceron, A., Yacef, K., ,(2003) "A Web-Based Tutoring Tool with Mining Facilities to Improve Learn- ing and Teaching." In Proc. of the 11th International Conference on Artificial Intelligence in Education. 12, IOS Press, pp. 201–208. IOS Press

[12] Merceron, A., Yacef, K., (2005) Interactive Multimedia Electronic Journal of ComputerEnhanced Learning. TADA-ED for Educational Data Mining. 12, pp. 1–13.

[13] Yacef, K., (2005) "The Logic-ITA in the classroom: a medium scxale experiment." AIED. 14, pp. 41–60.

[14] Pahl, C., Kennyand, C., (2009) "Interactive Correction and Recommendation for Computer Language Learning and Training." IEEE Transaction On Knowledge And Data Engineering. 21(6), pp. 854–866.

[15] Burton, R, Brown, J, (1976) "A tutoring and student modelling paradigm for gaming environments.", Proceedings of the SIGCSE-SIGCUE joint symposium on Computer science education. pp. 236–246.

[16] Khodeir, N., Wanas, N., Darwish3, N., Hegazy N., (2010) "Inferring the Differential Student Model in a Probabilistic Domain using Abduction Inference in Bayesian Networks." Proceedings of 3rd International Conference on Educational Data Mining Pittsburgh, PA, USA June 11-13, pp. 299–300.

[17] Carmona,C., Millan ,E., P´erez-de-la-Cruz, J., (2005) "Introducing Prerequisite Relations in a Multi- layered Bayesian Student Model." In: Ardissono, L. , Brna , P., Mitrovic, A. (eds.) User Modeling 2005, Springer-Verlag, pp. 347-356.

[18] Weber G., Brusilovsky, P., (2001) " ELM -ART: An Adaptive Versatile System for Web-based Instruction." International Journal of Artifical Intelligence in education, Vol. 12, pp. 351-384.

[19] Melis E., Siekmann, J., (2004) " ActiveMath: An Intelligent Tutoring System for Mathematics." Seventh International Conference 'Artificial Intelligence and Soft Computing' (ICAISC), In: Rutkowski and Siekmann J. , Tadeusiewicz R., and L. Zadeh (eds.) Lecture Notes in Artificial Intelligence Vol. 3070, Springer-Verlag, pp. 91-101.

[20] S. Suebnukarn, and P. Haddawy, (2005) " Modelling Individual and Collaborative Problem Solving in Medical Problem-Based Learning." In: Ardissono, L. , Brna , P., Mitrovic, A. (eds.) User Modeling 2005. Springer-Verlag, pp. 377-386.

[21] Nilsson, D., (1998) "An efficient algorithm for finding the M most probable configurations in probabilistic expert systems" Statistics and Computing. vol. 8, pp. 159–173

[22] The project Elvira, http://www.ia.uned.es/~elvira/index-en.html

[23] Collins, A., Greer, E., Huang, H. (1996) "Adaptive assessment using granularity hierarchies and Bayesian nets." In: Lecture Notes in Computer Science: Vol. 1086. Proceedings of 3rd International Conference ITS96, Berlin Heidelberg: Springer Verlag, pp. 569–577.

[24] Van Lehn, K., Niu, Z., Siler, S., Gertner, S., (1998) "Student modelling from conventional test data: A Bayesian approach without priors." In: Lecture Notes in Computer Science: Vol. 1452. Intelligent Tutoring Systems. Proceedings of 4th International Conference ITS98, Berlin Heidelberg: Springer Verlag, pp. 434–443.

[25] Mill´an, E., P´erez-De-La-Cruz, J. Luis., (2002) "A Bayesian Diagnostic Algorithm for Student Modelling and its Evaluation." User Modelling and User-Adapted Interaction. Vol. 12, 2-3, Kluwer Academic Publishers, Hingham, MA, USA. pp.281–330.

AUTHORS

Nabila Khodeir received Bachelor's in Electronics from Electronics Engineering at Monofia University and Master’s from Electronics and Communications department at Cairo University. She is PhD Student for computer s at Cairo University since 2006.

Nayer Wanas has been an Assistant Research Professor at the Electronics Research Institute in the Department of Informatics since 2003. He conducts research in the area of data mining. His main research interests are in data and text mining, information fusion and machine learning. Nayer has published over 30 technical papers in international journals and conference. In 2005 he was involved in the design and founding of the Data Mining and Computer Modelling Centre of Excellence in Egypt and held the position of program manager of the centre till 2007. He was also the founder and associate of knowledge for development, a consultancy in utilizing ICT for development. He was a postdoctoral fellow in the Pattern Analysis and Machine Intelligence lab at the University of Waterloo, Ontario, Canada where he focused on research in Multiple Classifier Systems. Since 2007 he remains to be affiliated with the Cairo Microsoft Innovation Laboratory (CMIC) where he is currently a Researcher conducting research in the area of mining social media and web 2.0 platforms. Nayer Wanas received his Ph.D. in Systems Design

Engineering from the University of Waterloo in 2003, and his Bachelors and ME from the Electronics and Communications department at Cairo University in 1992 and 1996 respectively.

Prof. Nevin Darwish is an active member of the IEEE and several other professional bodies. Her main scientific and research activities in the last ten years dealt with various aspects of Machine and Artificial Intelligence and Expert Systems, Natural Language Understanding, Intelligent Agents. Published more than 50 articles in national, international journals and conferences , was awarded the faculty graduate research excellence award for supervising best thesis and acted as an evaluator for main scientific Egyptian Journals as well as several national and international held conferences and funded projects by international intestates. Played the role of principle investigator as well as team member for several funded projects from the Academy of science and research. Currently is a team member of a Tempus supported project from with partners from Germany and France.

In addition to holding teaching positions at Cairo University, and teaching experience at McGill University, Canada, King Abdel Aziz University, Saudi Arabia and AUC-Cairo, Prof. Darwish acted on the board of directors of the Scientific Computation Center and Center of Programming and Biomedical Engineering, at Cairo University. She has also worked as a member of a consulting team to several national and international entities such as the Egyptian Television, and Unesco preparing and presenting desk research reports as well as being a member for technical tender evaluations. Prof. Nevin Darwish received her B.Sc. in Electronics and Communications Engineering 1974 and a second B.Sc. in Mathematics, Ain Shams University 1976. Next, she joined McGill University, Canada, and got her the M.Sc. in Computer 1978. Finally, she got the Ph.D. in Computer from Cairo University in 1983.

Prof. Nadia Hegazy received her Ph. D. in Computers from Cairo University. Her main research interests are in artificial Intelligence, Arabic computational linguistics, automatic translation, computers in education, multimedia systems, more than 25 years in language engineering. She was senior expert for strategic Planning since 2007, ministry of telecommunication and Information Technology, senior Expert for International Relations Ministry of telecommunications and Information Technology since 2002, professor of Computer Engineering since 1986 Electronics Research Institute, Head of the Informatics Department 1980-1994, and vice President of the Electronics Research Institute 1994-2001. She Supervised More than 45 M.Sc. and PH.D. theses and published more than 90 paper in the fields of her interest.