Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 50 (2016) 535 – 540

www.elsevier.com/locate/procedia

26th CIRP Design Conference

A BBN-based method to manage adaptive behavior of a smart user interface

Francesca Gullà[a]*, Lorenzo Cavalieri[a], Silvia Ceccacci[a], Michele Germani[a]

[a]Department of Industrial Engineering and Mathematical Sciences, Università Politecnica delle Marche, Via Brecce Bianche, 12-60131 Ancona, Italy

* Corresponding author. Tel.: +39-0712204880; *E-mail address:* f.gulla@univpm.it

Abstract

The present study proposes a new method to manage adaptation behaviour of adaptive system according to the output information provide by a user model based on Bayesian Belief Network (BBN). Such method has been applied in the development of smart interfaces for cooking and kitchen management, such as meal preparation and interaction with the major kitchen appliances, pandering the user's skills, expertise and disabilities. Nevertheless, this method is flexible and suitable enough to be used in other application contexts. The validity of the decision making algorithm has been tested through simulation of real user case scenarios.

Peer-review under responsibility of the organizing committee of the 26th CIRP Design Conference

Keywords: Adaptive Interface; Bayesian Network; Smart Home; Decision making algorithm

1. Introduction

Adaptive systems are entities that adapt their displays and available actions to the user's current goals and abilities, by monitoring user status, the system state and the current situation" [1]. To achieve this, such systems have to learn about the context in which they evolve, adapt their interfaces and interaction modalities used to communicate with users. Adaptive user interfaces may consider as a special class of learning systems that are developed to aid humans, in contrast with traditional ones, which was introduced with the aims to replace domain experts. [2].

One of the main feature that characterize an adaptive system is the user model. A user model is a representation of information about individual user. The amount and the nature of such information largely depend on the kind of system adaptation functionalities [3]. What seems to be quite specific for user modelling in adaptive systems is the sharing of the duties between the user and the systems in the user modelling process. This implies the importance of the right choice of the user modelling in terms of collecting user data, processing the data to build or update the correct model, and applying it to provide the desired adaptation. In order to manage the adaptation, depending on user model prevision, a good decision making strategy must be applied.

In the context of adaptive system development, the present study proposes a new method to manage adaptation behavior of adaptive system according to the output information provide by a user model based on Bayesian Belief Network (BBN). Such method has been applied in the development of smart interfaces for cooking and kitchen management, such as meal preparation and interaction with the major kitchen appliances, pandering the user's skills, expertise and disabilities. Nevertheless, this method is flexible and suitable enough to be used in other application contexts.

The validity of the decision making algorithm has been tested through simulation of real user case scenarios. The results highlight that the proposed decision making algorithm is able to effective readapt the interface in a reliable and efficient manner.

Peer-review under responsibility of the organizing committee of the 26th CIRP Design Conference
doi:10.1016/j.procir.2016.04.162

2. Research Background

User modelling is a domain in which there are many different sources of uncertainty and imprecision. In fact, as it is known, it's impossible to observe or modelling any phenomenon in a fully exhaustive and accurate way and consequently, a user model, based on previous user behaviour observation, can never certainty predict whether a user will behave in a certain way, at a given time. As is known, to manage models based on uncertain data, numerically-approximate reasoning techniques can be used [4]. The two methods most commonly applied in this context are the Bayesian Belief Networks and the Fuzzy Logic [3].

BBNs was introduced in the eighties [5]. Since then there has been an increasing interest, which led to the development of new techniques and algorithms and to an extension of models and applications. BBNs are a probabilistic model inspired by causality and provide a graphical model in which each node represents a variable and each link represents a causal influence relationship. These conditional dependencies represented by the graphical model estimated by using known statistical and computational methods. Hence, BBNs combine principles from graph theory, probability theory, computer science, and statistics [6].

The Fuzzy Logic paradigm was originally proposed in the sixties [7]. It is based on the idea that membership functions (in Fuzzy Sets theory) or truth values (in Fuzzy Logic) may take values between 0 and 1, with 0 representing absolute falseness and 1 absolute truth. Also FL has been the object of intense research, both in the development of models and in its application to real problems. Exhaustive review of fuzzy user modelling in the context of student modelling is presented in [8], while applications of this techniques in the context of adaptive web-based application are described in [9].

However, today the majority of researchers who decide to use approximate reasoning techniques in their user models choose the Bayesian network paradigm [10]. This is because they are one of the most complete and consistent knowledge tools for the acquisition, representation and use of data in conditions of uncertainty and they are considered one of the best available techniques for diagnosis and classification problems. They have emerged as a powerful technique for handling uncertainty in complex domains, and especially for context awareness to infer the context and provide reliable performance [11]. Moreover, they are well suited for use in the adaptive management for context-adaptive in a ubiquitous home environment principally because BBNs can learn from monitoring data [12].

The construction of BBNs consists of two steps:

- C*onceptual modelling*, which allow definition of BBN graphical structure made of nodes and link;
- *Definition of parameter* necessary to modelling data (e.g., conditional and prior probability or distributions).

In general, to collect information necessary to define BBN structure two method can be adopting: using domain expert's opinion or infer them from data using BBNs learning algorithms [13]. This second approach is the most powerful in order to develop automated user model, however, only few studies applied it to define both BBN structure and related parameters. [14]. In the majority of system described in literature structure is specified or constrained by experts and the parameters are partially or totally derived from data [15, 16, 17]. However, a key point of this approach is to choose a good value for the confidence level. If a too low value is selected, low data evidence relationships will be inferred. If the value is too high, some important relationships can be missed.

To solve this problem a proper decision making algorithm has to be defined.

3. The proposed method

A new method to manage a smart adaptive system, which is able to acquire knowledge about user and context through a set of physical and virtual sensors, is proposed. In particular, in order to detect user actions during interaction with the smart system interface, a "using" profile model is construct, which implement BBN to estimates the probabilities of the user expected behaviour. A decision making algorithm is defined to manage the correct adaptation action according to the user model output. Figure 1 shows an example of a general BN. In order to manage BBNs, we propose to use the simulation software Netica, produced by Norsys Software Corporation [18]. This software actually is a very usable tools for working with BBNs and influence diagrams. It easily allows to drawing Bayesian networks or to create them by using powerful Netica APIs toolkit, which make it interoperable with the major programming languages (i.e., Java, C, Visual Basic).

Moreover, it provides tools to:

- Detect optimal decisions for sequential decision problems and solve influence diagrams efficiently by using clique trees;
- Learn probabilistic relations from data, even with missing data;
- Allow the entry of probabilistic relations by equation, with an extensive built-in library of probabilistic functions and many other mathematical functions.

The user model is then design and implemented though the following steps:

1. User activity decomposition. A proper user activity decomposition analysis is carry out to identify: (a) actions performed by the user on the system, which are relevant to achieve specific goals; (b) system actions to detect user characteristics, which may affect user-system interaction (e.g., user's emotional, physiological and psychological states).
2. Context analysis. A context analysis is performed to understand how environment attributes can affect user-system interaction (e.g., year period, time of day, lighting condition, temperature, etc.).
3. Building of network structure in Netica. Definition of nodes representing context, user's actions, system actions and user's goals and define relations (i.e. flows) between each other.
4. Network modelling: for each node a collection of properties and states are defined.

5. Compilation of the Network. Compilation process allows Netica to build the junction tree relative to the inferences.
6. Network Learning and Simulation. It consists in learning probabilistic relations between random variables. To this purpose, Netica Software offered two different possibilities: manually enter the probabilistic relations or to import data to learn from existent case.

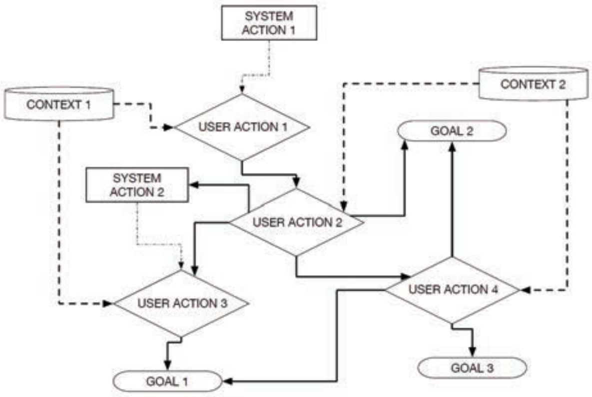

Figure 1. Example of a general BN

Netica automatically calculates the statistical probabilities of all events associated to each BN nodes. To allow the network to provide valuable support for identification of proper system adaptation action, a decision-making process has been defined, as described in the next sub-section.

To handle the multi-user situation at the same time a user profile is created for each user and BBN is populated by his/her interaction events.

3.1. *Definition of decision making algorithm*

In order to detect the appropriate system adaptive behaviour, a decision-making algorithm has been defined (DMA).

To be effective, DMA must support the identification of proper system adaptation actions, according to collected user and context data. To this end, one possible strategy is to verify the existence of a predominant probability for a specific event (e.g., user's action or goal) and trigger system adaptive action only in this case.

The DMA needs parameters able to describe features of the target stochastic distribution: two parameters are defined to achieve this purpose. Once defined the target distribution D, the first step of the algorithm is the detection of higher probability P_{max}. Two functional values can be computed according to the equations below.

$$\alpha = P_{max} - \frac{1-P_{target}}{N-1} \tag{1}$$

$$\rho = \sigma^2(D - P_{max}) \tag{2}$$

Where α describes the gap between the maximum probability and average residual probability: in this way, a first estimation of the dominance level is provided. On the other hand, ρ measures the standard deviation of the residual probability (all probability excluded the maximum): in this way, one can get information about how other probabilities are localized or distributed.

Then, it is necessary to define an algorithm able to provide in output a Boolean condition to inform the system about the right adaptation action: this process can be carried out defining appropriate thresholds for α and ρ. We respectively called these thresholds α_{th} and ρ_{th}. The adaptation event is triggered, if both the conditions $\alpha > \alpha_{th}$ and $\rho > \rho_{th}$ are verified. Otherwise, no adaptation is required.

The flowchart, shown in Figure 2, reports the described algorithm. In this reasoning a problem occurs: if the statistical distribution has only two levels, the standard deviation of residual probability will be surely zero.

Therefore, a by-pass routine is implemented to manage this exception: when 2-level distribution pass the first control, the adaptation action is directly performed.

The main issue to properly configure the algorithm in order to make it more effective is the choice of the two threshold values α_{th} and ρ_{th}: they are not default and must be defined empirically. Indeed, simulation batteries are carried out to find the best estimation of threshold values. Identifying proper estimation values is crucial for two reasons: on one hand, it allows to prevent adaptation determined by false positive events, which may result in continuous adaptations that could cause confusion in the user and, one the other hand, it avoids the occurrence of false negatives due to a too restrictive criterion.

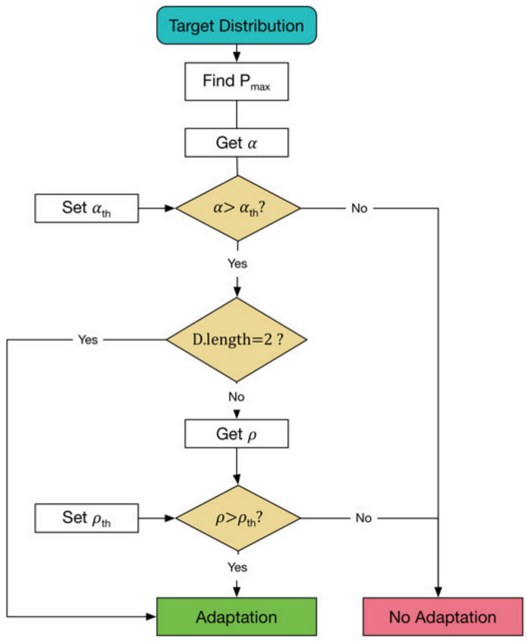

Figure 2: Decision Making Process Flowchart

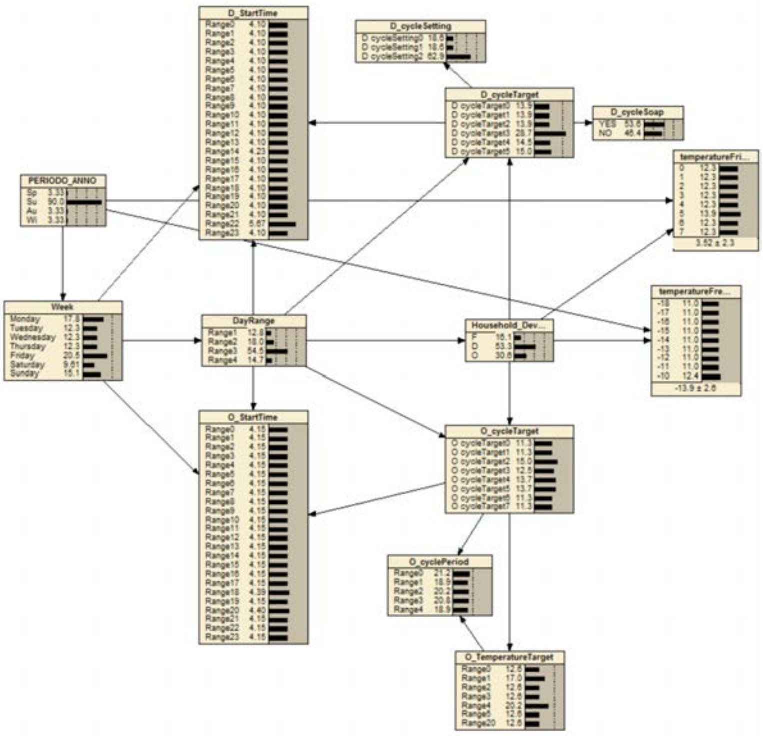

Figure 3. The resulted BBN

4. The case study

A BBN was built assuming the use of the system in a home environment; specifically, in the kitchen, characterized by three appliances: oven, dishwasher and refrigerator. The structure, shown in Figure 3, consists of 14 nodes, defined as:

1. Year_Period: to specify the four year periods (Fall, Winter, Spring and Summer).
2. Week: the seven days of the week (Monday Tuesday Thursday Friday Saturday and Sunday).
3. Day_Range: schedules the day in four time periods (range 1 = 06:01 to 12:00, Range2 = 12:01 to 18:00, Range3 = 18:01 to 12:00, Range4 = 0:01 to 6:00).
4. Household_Devices: characterized by three appliances: oven, dishwasher and refrigerator
5. O_CycleTarget: specifies the oven program setting.
6. O_CyclePeriod: is the oven set cooking times.
7. O_TemperatureTarget: Consists of the oven temperature setting.
8. O_StartTime: Involves the oven time setting.
9. TemperatureFreezer: the freezer temperature setting.
10. TemperatureFridge: the fridge temperature setting.
11. D_CycleTarget: the dishwasher program setting.
12. D_CycleSetting: the options set washing in the dishwasher
13. D_CycleSoap: the dishwasher options related to the presence / absence of detergent
14. D_StartTime: the dishwasher time setting.

The network links show the direct dependence between the nodes; for example, the network root node “Household_Devices” is directly influenced by three context nodes: year, day of week and time of day.

Depending on the appliances historical use, the node stochastic probability will be updated and return the more likely use devices in output. In turn this node will affect all the other nodes, which are related to oven, dishwasher and fridge data setting. Finally, this node will return the greater output use predictability. Once the network is built, it is necessary to set probabilistic relations between random variables.

To this purpose, existent case data from external database are automatically imported through the Learning Case function. Once the network was built, a simulation to evaluate the actual behaviour has been performed.

To create external database necessary to implement and tests adaptation functions, several use case scenarios have been defined for each target user profile. An example of the scenarios is reported in the Figure 4. An extract of the respective external database is reported in Figure 5. The database is designed to collect a set of structured data. In detail the columns represent the network nodes, the lines indicate the node status and are updated at each simulation with the system.

EXAMPLE OF SCENARIOS

John used the dishwasher once a day, usually in the evening, with the breakfast, lunch and dinner dishes. Frequently uses the Daily program serving detergent tablets for convenience. The weekend habits change: Sunday john sets the Intensive program and start washing immediately after lunch, using detergent tablets. Furthermore, John uses the oven 4 times a week; Mondays and Fridays for cooking the seafood he always sets a delayed start and uses the Ventilated program. On Saturday evening he uses the oven to cook pizza with Thermo-ventilated program. Every Sunday evening, he also activates the oven cleaning

Figure 4. Example of use case scenarios simulate to create learning data

Day	YEAR_PERIOD	Week	DayRange	Household_Devices	temperatureFridge	temperatureFreezer	O_cycleTarget	O_StartTime	O_TemperatureTarget	O_cyclePeriod	D_cycleTarget	D_StartTime	D_cycleSetting	D_cycleSoap
27/07/15	Su	Monday	Range2	F	5	-10								
28/07/15	Su	Tuesday	Range1	O			O_cycleTarget3	Range10	Range4					
28/07/15	Su	Tuesday	Range2	D							D_cycleTarget0	Range14	D_cycleSetting1	YES
28/07/15	Su	Tuesday	Range3	D							D_cycleTarget4	Range21	D_cycleSetting1	YES
29/07/15	Su	Wednesday	Range2	D			O_cycleTarget3	Range19	Range1					
29/07/15	Su	Wednesday	Range1	O						Range0	D_cycleTarget0	Range13	D_cycleSetting1	YES
29/07/15	Su	Wednesday	Range3	D							D_cycleTarget4	Range21	D_cycleSetting1	YES
30/07/15	Su	Thursday	Range3	O			O_cycleTarget5	Range19	Range5	Range0				
30/07/15	Su	Thursday	Range3	D							D_cycleTarget4	Range21	D_cycleSetting2	YES
31/07/15	Su	Friday	Range1	O			O_cycleTarget2	Range12	Range4	Range0				
31/07/15	Su	Friday	Range3	D							D_cycleTarget4	Range22	D_cycleSetting1	YES
01/08/15	Su	Saturday	Range1	O			O_cycleTarget3	Range10	Range4	Range0				
01/08/15	Su	Saturday	Range3	O			O_cycleTarget4	Range18	Range4	Range0				
01/08/15	Su	Saturday	Range2	D							D_cycleTarget0	Range15	D_cycleSetting1	YES
02/08/15	Su	Sunday	Range1	O			O_cycleTarget2	Range13	Range2	Range0				
02/08/15	Su	Sunday	Range3	O			O_cycleTarget3	Range20	Range5	Range0				
02/08/15	Su	Sunday	Range1	D							D_cycleTarget2	Range10	D_cycleSetting2	YES
02/08/15	Su	Sunday	Range3	D							D_cycleTarget2	Range21	D_cycleSetting2	YES
03/08/15	Su	Monday	Range2	O			O_cycleTarget2	Range12	Range2	Range0				
03/08/15	Su	Monday	Range3	O			O_cycleTarget4	Range20	Range5	Range0				
03/08/15	Su	Monday	Range3	D							D_cycleTarget4	Range21	D_cycleSetting1	YES
04/08/15	Su	Tuesday	Range1	O			O_cycleTarget3	Range19	Range1	Range0				
04/08/15	Su	Tuesday	Range3	D							D_cycleTarget4	Range21	D_cycleSetting1	YES
05/08/15	Su	Wednesday	Range3	O			O_cycleTarget2	Range13	Range20					
05/08/15	Su	Wednesday	Range2	O			O_cycleTarget3	Range13	Range5					
05/08/15	Su	Wednesday	Range3	D							D_cycleTarget2	Range21	D_cycleSetting1	YES
06/08/15	Su	Thursday	Range3	O			O_cycleTarget3	Range11	Range4					
06/08/15	Su	Thursday	Range3	D							D_cycleTarget4	Range21	D_cycleSetting2	YES

Figure 5. External database related Scenario described in Figure 4

5. Simulation and results

As previously described, the algorithm configuration needs two precise values of the threshold parameters (α_{th},$\rho_{th,}$). To reach this goal, a testing procedure is developed: the data management and processing is carried out through the Numerical Computation Software Scilab 5.2.2, using the script coding approach. The operational procedure is conducted as follows:

1. The Beliefs of a Test Bayes Network achieved with Netica Software are exported in a XML file and loaded on Scilab Environment;
2. Then, an empirical graphical evaluation allows to select the distributions suitable to trigger an adaptation action (Reference Distributions Numbers, RDN). Once chosen, the RDN can be defined as starting parameters of the simulation routine;
3. Then the variability ranges of the two threshold parameters, are chosen to define the simulation domain. In addition, the resolution (R) of the intervals is defined: it allows to determine how many values can be having the two ranges and it can be used to define the accuracy of the simulation. These mentioned parameters influence the number of simulations that can be made;
4. The filter processing provides the output as described by the DMA: for each simulation *i*, all the target distributions are filtered with the corresponding couple values ($\alpha_{th}^{(i)}$,$\rho_{th}^{(i)}$) and they are flagged if passing the filtering process;
5. In the last step, each simulation output (flag and unflag distributions) is compared with the reference solution defined in the first step. This operation allows to defined a subset of all simulation where the pass-filtered distributions match with the chosen distributions. Thus, the procedure provides a set of coupling values selectable to the algorithm configuration.

The simulation is performed on 14 target distribution, selecting five ones as described in step 2. The configuration parameters for the simulation procedure has been defined in Table 1: this configuration produces a simulation with 2500 samples: the comparison with the reference solution allows to obtain 294 couples of (αth,ρth) values suitable to perform the filter processing without the occurrences of false negatives or/and positives (Figure 6). The results show a rich set of combination for the two parameters. In order to select the best configuration, a battery of 1000 random distributions is performed varying the number of level from 2 up to 24. From this further test the selected threshold couple is defined (αth = 0,82 and ρth=0,57).

	Values
Reference Distributions Number (RDN)	5
Alpha Range	[0,1]
Rho Range	[0,2]
Resolution (R)	50

Table 1. Simulation Parameters.

The assessment allows to demonstrated the goodness of this approach as a tool for achieving a decision making algorithm based on the correlation between the probability value of the different level of a stochastic distribution.

The simulation used to obtain proper configuration parameters has been very effective, producing a consistent set of combinations. To refine the selection and reach a significant pair, the further simulation battery allowed to reach an output able to guarantee the perfect reliability compared to initial predictions, avoiding cases of false positive and/or negative.

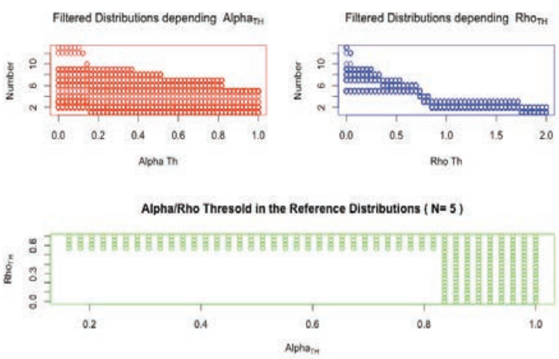

Figure 6. Simulation results.

6. Conclusion and Future Work

In the context of adaptive system development, this paper presents a method to manage adaptation behaviour of adaptive system according to the output information provide by a user model based on Bayesian Belief Network. Such method has been applied in the development of smart interfaces for cooking and kitchen management, such as meal preparation and interaction with the major kitchen appliances, according to the user's skills, expertise and disabilities.

In particular, in order to detect user actions during interaction with the smart system interface, a "user" profile model is construct, which implement BBN to estimates the probabilities of the user expected behaviour. In order to manage BBNs, we propose to use the simulation software Netica, produced by Norsys Software Corporation.

A BBN was built assuming the use of the system in a home environment; specifically, in the kitchen, characterized by three appliances: oven, dishwasher and refrigerator. Once the network was built, a simulation to evaluate the actual behaviour has been performed. To allow the network to provide valuable support for the identification of a proper system adaptation action, a decision-making algorithm has been defined. To be effective, DMA must support the identification of proper system adaptation actions, according to collected user and context data. To this end, one possible strategy is to verify the existence of a predominant probability for a specific event (e.g., user's action or goal) and trigger system adaptive action only in this case.

The validity of the decision making algorithm has been tested through simulation of real use case scenarios. The results highlight that the proposed decision making algorithm is able to readapt the interface in a reliable and efficient manner.

The proposed method can be useful to manage any service or system based on Internet of Things paradigm in several contexts that range from Smart Factory to medical applications.

For example, in the smart factory context, it can be useful to implement a system able to support numerical control machines settings according to design and production issues. Otherwise, in medical field it may allow to improve the management of therapy.

Acknowledgements

This work has been developed in the context of "D4All: Design for all" project, National Technological Cluster funded by the Italian Minister of University and Research.

References

[1] Rothrock, L., Koubek, R., Fuchs, F., Haas, M., & Salvendy, G. (2002). Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms. Theoretical Issues in Ergonomics Science, 3(1), 47-84.

[2] Langley, P. (1999). User modeling in adaptive interfaces. In Proc. Seventh International Conference on User Modeling (pp. 357-370).

[3] Brusilovsky, P., & Millán, E. (2007, January). User models for adaptive hypermedia and adaptive educational systems. In *The adaptive web* (pp. 3-53). Springer-Verlag.

[4] Zukerman, I., & Albrecht, D. W. (2001). Predictive statistical models for user modeling. User Modeling and User-Adapted Interaction, 11(1-2), 5-18.

[5] Pearl, J.: Probabilistic Reasoning in Expert Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc, San Francisco (1988)

[6] Nyberg, J. Brian, Bruce G. Marcot, and Randy Sulyma. "Using Bayesian belief networks in adaptive management." Canadian Journal of Forest Research 36.12 (2006): 3104-3116.

[7] Zadeh, L.: Fuzzy sets. Information and Control 8 (1965) 338-353

[8] Jameson, A.: Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues. User Modeling and User-Adapted Interaction 5 (1996) 193-251.

[9] Kavcic, A.: Fuzzy user modeling for adaptation in educational hypermedia. IEEE Transactions on Systems, Man, and Cybernetics 34, 4 (2004) 439-449.

[10] Brusilovski, P., Kobsa, A., & Nejdl, W. (Eds.). (2007). The adaptive web: methods and strategies of web personalization (Vol. 4321). Springer Science & Business Media.

[11] Dey, A. K. (2001). Understanding and using context. Personal and UbiquitousComputing, 5, 20–24.

[12] Song, In-Jee, and Sung-Bae Cho. "Bayesian and behavior networks for context-adaptive user interface in a ubiquitous home environment." Expert Systems with Applications 40.5 (2013): 1827-1838.

[13] Heckerman, D.: A Tutorial on Learning with Bayesian Networks, Technical Report No. MSR-TR-95-06, Microsoft Research Advanced Technology Division (1995)

[14] Mayo, M., & Mitrovic, A. (2001). Optimising ITS behaviour with Bayesian networks and decision theory. International Journal of Artificial Intelligence in Education, 12(2), 124-153.

[15] Millán, E., Agosta, J.M., Perez de la Cruz, J.-L.: Bayesian Student Modelling and theProblem of Parameter Specification. British Journal of Educational Technology 32, 2 (2001) 171-181

[16] Millán, E., Perez de la Cruz, J.-L.: A Bayesian Diagnostic Algorithm for Student Modeling. User Modeling and User-Adapted Interaction 12 (2002) 281-330.

[17] Vomlel, J.: Bayesian Networks in Educational Testing. International Journal of Uncertainty, Fuzziness and Knowledge-Based System 12 (2004) 83-100

[18] Netica Application. A complete software Package to solve problem using Bayesian Belief Networks and Influence Diagrams. Norsys Software Corp