Distributed Bayesian Networks for User Modeling

Roberto Tedesco	Peter Dolog	Wolfgang Nejdl	Heidrun Allert
Politecnico di Milano - DEI	Aalborg University	L3S Research Centre	FH-OO Studienbetriebs GmbH
Italy	Denmark	Germany	Austria
tedesco@elet.polimi.it	dolog@cs.aau.dk	nejdl@l3s.de	Heidrun.Allert@fh-hagenberg.at

Abstract: The World Wide Web is a popular platform for providing eLearning applications to a wide spectrum of users. However – as users differ in their preferences, background, requirements, and goals – applications should provide personalization mechanisms. In the Web context, user models used by such adaptive applications are often partial fragments of an overall user model. The fragments have then to be collected and merged into a global user profile. In this paper we investigate and present algorithms able to cope with distributed, fragmented user models – based on Bayesian Networks – in the context of Web-based eLearning platforms. The scenario we are tackling assumes learners who use several systems over time, which are able to create partial Bayesian Networks for user models based on the local system context. In particular, we focus on how to merge these partial user models. Our merge mechanism efficiently combines distributed learner models without the need to exchange internal structure of local Bayesian networks, nor local evidence between the involved platforms.

Introduction

Obviously users differ in their preferences, background, requirements, and goals they have when exploring instructional materials. This challenges the application to appropriately handle different users and situations. Personalization of information is a possible solution for handling these diverse needs, and quite a few adaptive applications have been developed in areas like eCommerce (Ardissono et al. 2000), eLearning (Dolog et al. 2004, Dolog et al. 2004a), or office applications (Oliver et al. 2005).

In eLearning, cognitive models can characterize the learner's ability to process information in a certain way and classify them to certain categories with respect to their information processing abilities and preferences. The cognitive model is used to adjust navigation guidance for a learner in the learning path proposed by the eLearning platform.

In the Web context, user models, constructed and used in adaptive applications, are often partial fragments of a *overall* user model. For example, let us imagine that Katia is a student interested in several domains. Katia attends several on-line courses provided by different Learning Management Systems (LMS's). Each LMS comes with its own user model and therefore has a different "view" on Katia's characteristics. To combine the observations and to make personalization more effective, we need to integrate the conclusions about Katia's learner characteristics and preferences from several systems.

Bayesian Networks are particularly suited for model integration. In our setting we therefore investigate the integration of distributed, fragmented user models by means on Bayesian Networks. Each Bayesian Network is created by a different application, yet using a common vocabulary regarding learning characteristics and preferences. Our model does not require the exchange of detailed user information like user logs.

The rest of the paper is structured as follows. The first section presents some works in the field of learning modeling and introduces the Dunn & Dunn model. The second section describes Bayesian Networks and their implementation in the context of the Virtual Campus project. The third section introduces the methodology we employ to merge learners' models. The fourth section discusses some related work about merging of Bayesian Network models. Finally, the last section draws some conclusions.

Modeling users' preferences -- Learning Styles

Given the same conditions and resources for learning, individuals perform differently and achieve different learning outcomes. This is mainly ascribed to unequal previous knowledge, motivation and intellectual properties. Several learning models are based on the concept of *cognitive style*.

The construct of cognitive styles relies on the assumption that styles of information processing (i.e. perceiving, attention, generating hypotheses in problem solving) are stable across various situations and can not be attributed to intelligence only. Thus, cognitive styles are an important part of a learner's personality (Oerter 71). They result from schematizing effective strategies of learning and problem solving. They are stable individual preferences for information processing, derived from generalizing effective strategies (Seel 2003).

There is no definite set of cognitive styles agreed on in psychology. Klix distinguishes *global strategies* from *local strategies* in problem solving (Klix 71). The local strategy is to produce partial solutions, testing them sequentially. The global strategy descends from the solution and solves the problem effectively. Witkin and Goodenough (Witkin et al. 81) distinguish *field-dependent* from *field-independent* cognitive styles. Field-dependent learners tend to accept facts and issues as presented. They find it hard to localize information in a complex environment and appreciate guidance and social interaction. Field-independent learners tend to reorganize and restructure information on their own. They need less guidance and are less geared to social interaction.

Messnick (Messnick 94) distinguishes several constructs relevant in learning and information processing: cognitive style (relevant in perceiving and thinking), learning style (preferences with regard to learning strategies), expressive style (relevant in communication), reactive style (self-reflection), defensive style (managing conflicts and anxiety), styles of cognitive control (metacognition).

Weinstein (Weinstein et al. 86) lists *strategies of elaboration* (relating the new to the previously learned by using examples, analogies, visualization techniques etc., and *strategies of organizing new information* (semantic classifying, using diagrams, etc.)

It is important to note that such models are scientific constructs. It is neither possible to directly measure and observe a cognitive style nor it is possible to deduce explicit consequences. The task to identify a cognitive style raises crucial questions. There are several inventories available, though there is no evidence whether they actually measure the construct of cognitive styles.

In this paper we borrow from the Dunn & Dunn model of *Learning Styles* (Dunn & Dunn, 78). The model focuses on instructional and environmental preferences. In particular, Dunn defines a learning style as "the way in which each learner begins to concentrate on, process, and retain new and difficult information". The model comprises five major categories called *stimuli*: Environmental, Emotional, Sociological, Physical, Psychological. Within these five major categories are 21 different elements that influence our learning, for example learning in pairs or from peers in the sociological category, or perceptual strengths in the physical category.

We assume that the adaptive system can deduce learning style characteristics from user interaction with the system, e.g. from observations about solved exercises, number of chat messages, number of comments for learner portfolios, and so on.

Bayesian User Models

Several techniques can be exploited to model users' preferences. Logics, Fuzzy Logic, Neural Networks, and statistical models represent some examples. Statistical models fit very well to the problem of modeling users, since they allow us to represent the intrinsic uncertainty inevitably related to any effort to model human characteristics. Bayesian Networks (BNs) in particular provide us with a simple yet effective approach to construct and handle statistical models (Pearl 88). Bayesian Networks have been employed to derive learner characteristics in several systems already (Sbattella et al. 2004), or to infer user goals (Horvitz et al. 98).

Bayesian Networks

BNs allow for a simple graphical representation of a multi-dimensional joint distribution P(A_1, A_2,… A_n). The model is a digraph composed of *nodes*, *oriented arcs*, and *conditional probability distributions*. Nodes represent stochastic variables that encode events. As usual, each variable takes values in a set defining the admissible states of the related event. An arc coming from node *A* and entering node *B* states that *A* is a *direct cause* of *B* (and node *A* is then said to be a *parent* or *B*). Each node which has at least one parent is associated with a *conditional probability distribution* (CPD) which encodes the probabilistic dependence P(*A* | parents(*A*)). Each node *R* with no parents (*root* nodes) is associated with its *prior probability distribution* P(*R*). Thanks to the fact that only direct causes need to be explicitly encoded by CPDs, BNs represent joint probability distribution in a concise and efficient way. If variables are discrete, distributions reduce to *conditional probability tables* (CPTs), and *prior probability tables*.

In a typical BN, a subset of the nodes represents observable events, i.e. events that can be collected from the environment and assigned to nodes as new *evidence* (or *fact*). The inference algorithm, starting from observed evidence, calculates the probability related to all non-observed nodes. In particular, leaf nodes are often used to collect events, while root nodes represent the *causes*. This is called *diagnostic reasoning*: From facts, we reconstruct the probability of causes.

Fig. 1 depicts the "laboratory session" example, in which we use discrete variables that take values in the {**T**rue, **F**alse} set. The variables represent probability about the following statements: "the learner likes laboratory sessions" (the *Laboratory* variable), "the learner tends to work with peers during laboratory sessions" (the *peerS* variable), "the learner prefers hands-on sessions" (the *pErceptual* variable), and "the learner sends several chat messages to the same person" (the *ChatMsg* variable). For example, the prior probability table associated to *Laboratory* states that the probability for a given learner to like laboratory sessions is 60%. This value comes from a-priori knowledge we suppose to have, about the distribution of the event we model by means of the stochastic variable *Laboratory*.

CPTs encode dependencies. For example, CPT associated to *ChatMsg* states that the probability for a given learner to send several messages to the same person, given that she/he likes to work in a group and does not like hand-on sessions, is 70%.

The tables can be filled by an expert or can be learned by means of a training algorithm. Given the model, suppose that, at run-time, we know that learner L exchanged a lot of messages during the laboratory session. We assign "T" to the *ChatMsg* node. The inference algorithm updates the distribution of Laboratory and build distributions for peers and perceptual. These are called *posterior probabilities* (see Fig. 1, on the right).

As a final remark, notice that only root nodes can be assigned with prior probabilities. However it can be the case that we have a-priori knowledge on the distribution of some non-root node. In this case, a well-known technique called *soft evidence* can be exploited to simulate prior probabilities on non-root nodes. In the following we assume the use of this soft-evidence technique whenever we need to assign prior probabilities to non-root nodes.

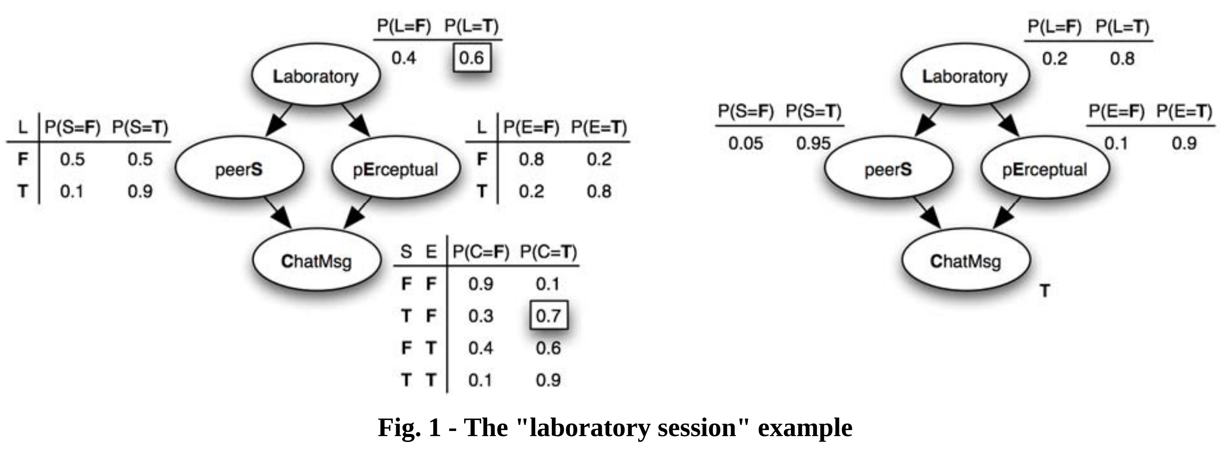

Fig. 1 - The "laboratory session" example

Virtual Campus Bayesian User Models

Our Virtual Campus Project implements a recommendation system based on the aforementioned techniques. Virtual Campus is a LMS developed at Politecnico di Milano, described in more detail in (Cesarini et al. 2004). In this context we implemented a Profile Engine (PEngine), which exploits the BN paradigm using discrete variables that take values in {Low, Medium, High}.

As users change their behavior over time, PEngine extends the BN model exploiting techniques from Dynamic BNs (Norvig & Russell 95). Whenever a learning activity ends, new evidence is asserted on observable nodes and posterior probabilities are calculated. These posterior probabilities become the new prior probabilities, for the same BN, when a new learning activity begins.

Fig. 2 depicts a simplified architecture of PEngine. Data is collected from Virtual Campus application logs. Since PEngine exploits a discrete BN, data must be discretized by means of a clustering algorithm. Then, new (discrete) evidence enters the BN, as well as new prior probability values coming from the previous evaluation cycle. The resulting new posterior probability values represent the new state of the user model. The Virtual Campus Tutoring Module exploits the model to generate personalized advice.

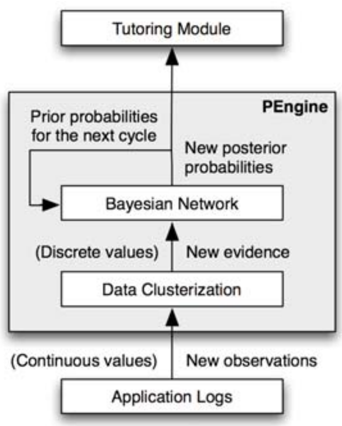

Fig. 2 - PEngine architecture

Distributed User Models

In our scenario, users use different eLearning platforms, each one with its own user model. If we do not make any assumptions on these user models, it is impossible to exchange knowledge the systems have gained about their users, due to both differences in standard and modeling techniques, and unsolved issues in model composition.

Bayesian Network Chains

In this paper we assume all the involved system to adopt a BN-based user model, exploiting discrete variables. We also assume that all BNs use a common vocabulary for learner characteristics, given by a learning style model such as the one sketched earlier in this paper. We can then investigate model composition: How to reuse partial knowledge about users to build a consistent global user model.

In our scenario, Katia attends on-line courses provided through different LMS's, switching from one to another whenever she does not find the instructional material she is looking for. Fig. 3 depicts three of these systems (called LMS_1, LMS_2, and LMS_3), each one coming with its own fragment of Katia's user model.

The BNs in our example implement a small subset of the D&D model elements. Leaf nodes represent information gathered by the system from learner activities, upper nodes implement the D&D elements. CPTs encode the effects of these elements on learner actions. The standard BN inference algorithm will compute the probability of D&D elements starting from collected data

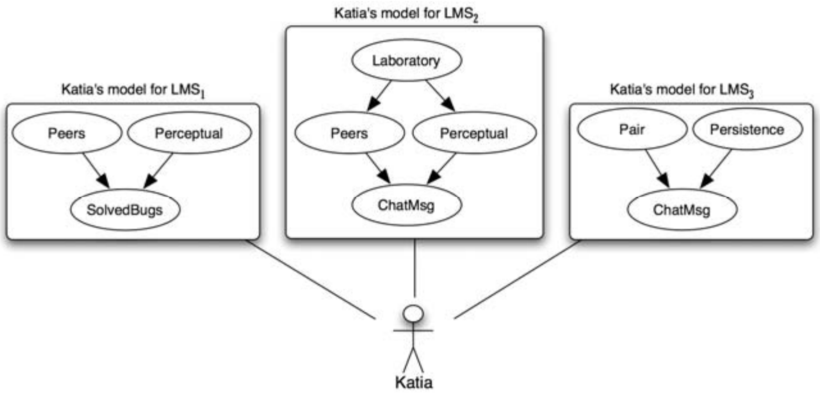

Fig. 3 - Three partial models about Katia

To merge user model fragments, we have to extend the PEngine model depicted so far. We will assume that different BNs could have different structures, CPTs, prior probability values, and even different *structure*. As a result we obtain a *chain* of different BNs: A BN for each different platform involved in modeling the user. Fig. 4 depicts an example in which three LMS's, and therefore three different user model fragments, are involved. During this instructional path, the learner accesses LMS's in the following order: LMS_1, LMS_2, LMS_3, and again LMS_1.

Dashed arcs represent prior probabilities, while dotted arcs represent posterior probability. Posterior probability calculated for *shared* nodes (i.e. nodes that share the same semantics) are passed among the platforms. For example, LMS_2 receives *Peers* and *Perceptual* from LMS_1. In particular, as *Peers* and *Perceptual* are not root nodes in the LMS_2 BN, the probabilities are assigned as soft evidence. Root nodes that are *private* (i.e. they do not share the same semantics) are assigned a default prior probability value of (1/3, 1/3, 1/3). The main advantage of this approach is that there is no need to build a globally shared network. Instead, each LMS still evaluates only its own BN and does not have to share or send any evidence/CPTs to other LMS's. The algorithm performed by LMS_i at time T_j is sketched here:

```
Gather message M_{Tj-1} coming from LMS_{k,Tj-1}
For each node X of LMS_{i,Tj} BN:
    If X ∈ M_{Tj-1} Then
        Extract P(X) from M_{Tj-1}
        If X is root node on LMS_{i,Tj} BN Then
            Assert P(X) as prior probability
        Else
            Assert P(X) as soft evidence
        Endif
    Else
        If X is not root node on LMS_{i,Tj} BN Then
            Assert (1/3, 1/3, 1/3) as prior probability
        Endif
    Endif
Endfor
Assert hard evidence on observed nodes of LMS_{i,Tj} BN and compute new posteriors
```

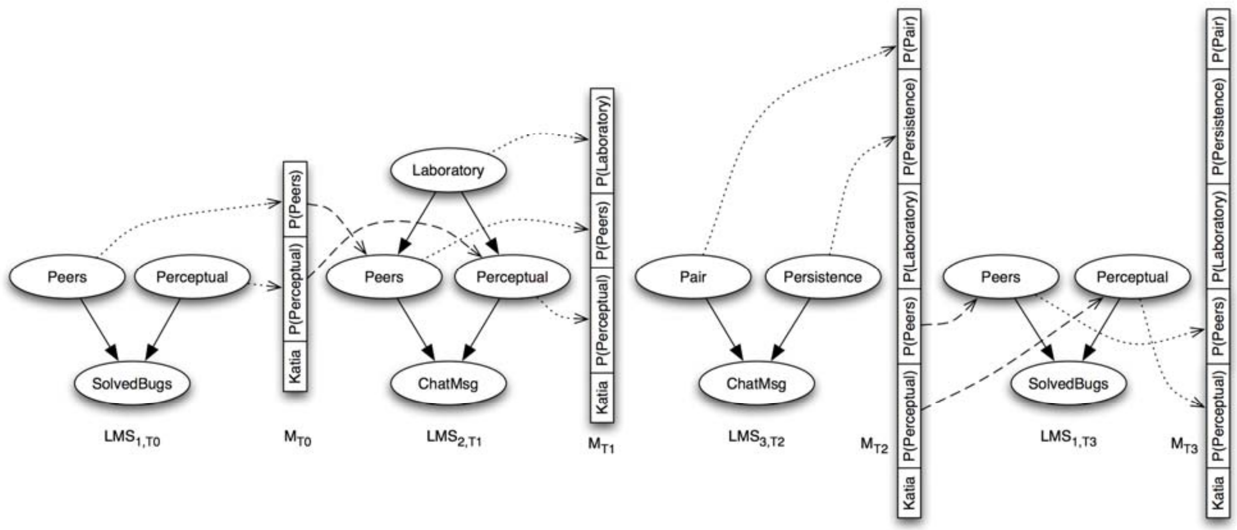

Fig. 4 – A chain of Bayesian Networks

An example -- The Katia's learning path

In our example, Katia begins her learning path exploiting on LMS_1, attending a collaborative laboratory session on programming languages. The BN tries to derive Katia's preferences about team-based vs. individual learning (node *Peers*) and visual/auditory materials vs. kinesthetic involvement (node *Perceptual*). The intended meaning is that if *Peers* is High, Katia is likely to prefer team-based learning style; if *Perceptual* is High, Katia tends to prefer activities that involve making things (i.e., science projects, storybooks, diaries, model building, etc). At the end of the laboratory session the system, relying on the number of software bugs solved by Katia during the session, updates the probability distribution for *Peers* and *Perceptual*.

Then, Katia leaves LMS_1 and enters a collaborative session on modern art on LMS_2. LMS_2 has some knowledge about *Peers* and *Perceptual* coming from LMS_1, and exploits it asserting soft evidence on related nodes in its BN. At the end of activity on LMS_2, the system, relying on the number of chat messages exchanged during the session, updates *Peers*, *Perceptual*, and *Laboratory*. Now our belief about Katia's abilities is changed. Moreover, we know another of Katia's characteristics: *Laboratory* (i.e. the ability of Katia to take advantage of laboratory sessions). Notice that *Laboratory* is not actually part of the D&D model: It represents a further learner characteristic, built upon the standard D&D elements.

As Fig. 4 depicts, Katia selects another collaborative learning experience, hosted on LMS_3. Node *Pair* is High if Katia tends to work with one other student, as opposed to be part of a larger group, while exploiting collaborative instructional materials. Node *Persistence* indicates Katia's ability to “stay on task” (preference for a single thread of discussion at a time). Notice that prior probability values of BN root nodes remain at initialization values, as their semantics do not match with the meaning of other already-computed nodes. At the end of fruition on LMS_3, the system, relying on the number of chat messages sent during the lesson, evaluates posterior probability values for *Pair* and *Persistence* and we have knowledge about two additional Katia’s characteristics.

As a final step Katia returns to LMS_1 and enters another collaborative laboratory session. The BN nodes are given with prior probability coming from LMS_2. At the end of activity on LMS_1, the system, relying on the number of solved bugs, updates posterior probability values for *Peers* and *Perceptual*.

As a final comment, notice that neither information about structure and CPTs of local BNs, nor local evidence has been exchanged among the LMS’s. This way, we avoid sharing sensible information about both the system and the learner. Moreover, we minimize the exchange of data.

Related Work

Some work can be found focusing on evaluation of a unique Bayesian Network shared among several systems. The network is partitioned into several subnetworks and each system is in charge of evaluating a specific subset of root nodes. Whenever a given system needs to compute its root nodes, the algorithm start in all the involved systems and calculates results from scratch. In the following we discuss some approaches.

The problem of calculation of root nodes, relying on local evidence and as few as possible evidence from other agents, is investigated in (Shen et al. 2002). The goal of the paper is to provide a technique able to reach a given confidence level, minimizing communications of evidence. The solution is based on the fact that, knowing CPTs, it is possible to calculate P(*unknown_evidence* | *known_evidence*). The solution is refined further in (Shen et al. 2003), using a Decentralized Markov Decision Process (DMPD) to calculate the best communication strategy. Both papers assume a simple, two-level BN. Structure and CPTs are supposed to be common knowledge.

A different scenario is faced in (Paskin et al. 2004): The network structure and CPTs are no longer common knowledge. The network structure is again a two-layer BN, and CPTs can be exchanged among agents. The paper proposes a distributed version of the well-known message-passing algorithm, which takes into account possible loss of data during the message-passing phase.

All the aforementioned papers start from a unique BN and, in different ways, try to evaluate it in a distributed manner. These approaches need several messages to be exchanged among the involved systems, communicating data during the distributed evaluation process. Moreover, these approaches work efficiently with simple two-layer BNs.

Instead, our approach starts from several *different* BNs, connect them as a chain and reuse posterior probability, coming from the previous BN, as prior probability for the current BN. This way, only minimal information need to be exchanged among the involved systems. Moreover, our approach works efficiently with multi-layer BNs.

Conclusions and Future Work

Providing personalization capabilities to eLearning applications promises increased user satisfaction and system effectiveness. Appropriate user models can be inferred from user interactions with LMS's. In most cases, the user interacts with several systems in different ways, and the user models in these systems can then be considered fragments of a larger complete user model.

In this paper, we discussed how to represent user models based on learning style models (such as the D&D pedagogical model) as well as the inference of the relevant user characteristics from user interactions as Bayesian Networks. We then provided an algorithm to merge distributed user model fragments represented as Bayesian Networks, without the need to exchange details of local models nor local evidences. Our approach makes merging these models both efficient and, as much as possible, privacy-preserving.

One interesting issue we want to focus on in future work is to investigate scenarios where communication failures can occur between the systems providing the distributed user model fragments, making approximation of unknown values necessary. Second, we will investigate the use of ontology mapping techniques to cope with more than one vocabulary used for the distributed user model fragments.

References

Ardissono, L., & Goy, A. (2000). Tailoring the Interaction with Users in Web Stores. *User Modeling and User-Adapted Interaction*, 10 (4), 251-303.

Cesarini, M., Guinea, S., Sbattella, L., & Tedesco, R. (2004). Innovative learning and teaching scenarios in Virtual Campus. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2004)*, Lugano, Switzerland.

Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004). The personal reader: Personalizing and enriching learning resource using semantic web technologies. *Proceedings of AH2004 - International Conference on Adaptive Hypermedia*, Eindhoven, The Netherlands.

Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004a). Personalization in Distributed eLearning Environments. *Proceedings of WWW2004 - International World Wide Web Conference*, New York, USA.

Dunn, R., & Dunn, K. (1978). *Teaching students through their individual learning styles: A practical approach*. Englewood Cliffs, NJ: Prentice Hall.

Horvitz, E., Breese, J. S., Heckerman, D., Hovel, D., & Rommelse, K. (1998). The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. *Proceedings of UAI '98: the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, Wisconsin, USA.

Klix, F. (1971). *Information und Verhalten*. Bern: Huber.

Messnick, S. (1994). The matter of style: Manifestions of personality in cognition, learning, and teaching. *Educational Psychologist*, 29, 121-136.

Norvig, P., & S. J. Russell, S. J. (1995). *Artificial intelligence -- A modern approach*. Prentice-Hall.

Oerter, R. (1971). *Psychologie des Denkens*. Donauwörth: Auer.

Oliver, N., & Horvitz, E. (2005). A Comparison of HMMs and Dynamic Bayesian Networks for Recognizing Office Activities. *Proceedings of User Modeling 2005: 10th International Conference*, Edinburgh, Scotland, UK.

Paskin, M., & Guestrin, C. (2004). Robust Probabilistic Inference in Distributed Systems. *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*, Arlington, Virginia, 436-445.

Pearl, J. (1988). Probabilistic reasoning in intelligent systems -- Networks of plausible inference. San Mateo, CA: Morgan Kaufman.

Sbattella, L., & Tedesco, R. (2004). Profiling and tutoring users in Virtual Campus. *Proceedings of the 5th International Conference on Information Technology Based Higher Education and Training (ITHET '04)*, Istanbul, Turkey.

Seel, N. M. (2003). *Psychologie des Lernens*. Stuttgard: UTB.

Shen, J., Lesser, V., & Carver, N. (2002). Controlling Information Exchange in Distributed Bayesian Networks. *UMASS Tech Report 02-22*.

Shen, J., Lesser, V., & Carver, N. (2003). Minimizing Communication Cost in a Distributed Bayesian Network using a Decentralized MDP. *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, Melbourne, AUS, 678-685.

Weinstein, C. F., & Mayer, R. F. (1986). The teaching of learning styles. *Handbook of research on teaching. 3rd edition*. New York: Macmillan.

Witkin, H. A., & Goodenough, D. R. (1981). *Cognitive styles: Essence and origins: Field dependence and independence*. New York: International Universities Press, Inc.