DESIGNING INTELLIGENT TUTORING SYSTEMS: A BAYESIAN APPROACH

Hugo Gamboa
Escola Superior de Tecnologia de Setúbal, Campo do IPS, Estefanilha, Setúbal Portugal
Email: hgamboa@est.ips.pt

Ana Fred
Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa Portugal
Email: afred@lx.it.pt

Keywords: Intelligent Tutoring Systems, Bayesian Networks, e-Learning.

Abstract: This paper proposes a model and an architecture for designing intelligent tutoring system using Bayesian Networks. The design model of an intelligent tutoring system is directed towards the separation between the domain knowledge and the tutor shell. The architecture is composed by a user model, a knowledge base, an adaptation module, a pedagogical module and a presentation module. Bayesian Networks are used to assess user's state of knowledge and preferences, in order to suggest pedagogical options and recommend future steps in the tutor. The proposed architecture is implemented in the Internet, enabling its use as an *e-learning* tool. An example of an intelligent tutoring system is shown for illustration purposes.

1 INTRODUCTION

Intelligent tutoring systems (ITSs) are software programs that give support to the learning activity. These systems can be used in the normal educational process, in distant learning courses, either under the form of CDROMs or as applications that deliver knowledge over the Internet. They present new ways for education, which can change the role of the human tutor or teacher, and enhance it.

Computers have been used in education sincefor over 25 years (Beck, Stern, & Haugsjaa 1996). First developments towards the introduction of software programs as auxiliary tools in the learning process were formalized in the area of computer aided instruction (CAI). These programs used a decision tree to guide the student from one content page to another depending on his/her answers (Urban-Lurain 1996). These programs didn't consider the diversity of students and their particular needs, background or history. CAI programs failed to adapt to the specific way of acquiring knowledge a student had, and couldn't give the individualized attention that a human tutor provides (Bennett 1997).

Recent research developments in Artificial Intelligence led to a new application field, named Intelligent Tutoring Systems (Burns & Capps 1988). The basic underlying perspective is to consider every student as being unique. This paradigm is based on the creation of a user or student model (VanLehn 1988) that keeps records on user preferences and progress during the knowledge acquisition process.

The main characteristic that increases effectiveness to the task of teaching with an ITS, is the adaptability to the student. Assessing the user state of knowledge and profile requires uncertainty reasoning. Artificial Intelligence has addressed this problem in various ways (Nilson 1998) such as certainty factors in knowledge representation (Buchanan & Shortliffe 1985), probability theory or fuzzy logic (Zimmerman 1991). Bayesian Networks (BNs) (Pearl 1998)(Frey 1998) are a most actual approach to uncertainty modelling. This technique combines the rigorous probabilistic formalism with a graphical representation and efficient inference mechanisms (Jensen 1997).

Examples of ITSs (Brusilovsky, Schwarz, & Weber 1996)(Frölich & Nejdl 1997)(Schulze *et al.* 2000) present various ways of dealing with the problem of tutoring using the computer, suggesting different approaches to the user model.

A typical ITS architecture separates domain application dependent parts—including a knowledge base which stores course contents—from generic parts which are independent of the knowledge to be taught. The same approach has been adopted in the field of expert systems (Buchanan & Shortliffe 1985)(Fred 1994)(Russel & Norvig 1995), according to which the set of all domain independent parts of the system is called a *shell* (an empty knowledge base, inference mechanisms and eventually a design interface). This generic part in intelligent tutoring systems is sometimes called *authoring tool*. This structure is being used to accelerate the developing process of new applications (Fleming & Horwitz 1996)(Murray 1999).

The potentialities that the Internet provides, makes it the best place to locate an ITS for distance education or as an *e-learning* tool (Ubell 2000). By scanning the literature on ITS, one can identify three main classes according to the systems main goal (Murray 1999): (a) problem solving guidance (Schulze *et al.* 2000); (b) curriculum sequencing (Brusilovsky, Schwarz, & Weber 1996); (c) device/process simulation. No global solution has yet been proposed covering the several aspects.

In this paper we propose a model and an architecture for the creation of ITSs under a Bayesian approach supported.

The global model is decomposed in interactive submodels explicity addressing user modelling, knowledge design and Bayesian inference. This approach provides modularity to create tutors in different contexts with the same architecture. The shell, the part of the tutor that is teaching domain independent, is divided into five parts: user model, knowledge model, adaptation module, pedagogical module and presentation module.

The ITS shows the contents to be learned via an Internet browser that, through a set of content specialized pages, interacts with the student and suggests pedagogical options. A Bayesian Network combines the contents structure with the user profile and learning style to suggest pedagogical directions.

Section 2 gives a brief introduction to BNs in the context of user modelling; in section 3 the design model for ITSs is proposed; section 4 describes the architecture; implementation aspects are treated in section 5; and section 6 illustrates the use of the proposed architecture and model a draft of an ITS ia a topic of *Pattern Recognition*.

2 BAYESIAN NETWORKS

A Bayesian Network is a graphical description of a probability distribution that permits efficient probability propagation combined with a rigorous formalism. A BN for a given domain represents the joint probability distribution, $p(x)$, over the set of random variables, X, of the domain, as a set of local distributions combined with a set of conditional independence assertions. This joint probability is computed from expression 1,

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \Pi_i) \qquad (1)$$

where, for each variable x_i, $\Pi_i \subseteq \{x_1, \ldots, x_{i-1}\}$ is a set of variables of which x_i is conditionally dependent.

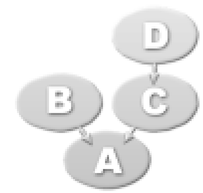

Figure 1: A Bayesian Network.

The BN structure encodes the assertions of conditional independence as a directed acyclic graph such that: (a) each node corresponds to a variable; (b) the parents of the node corresponding to x_i are the nodes associated to the variables in Π_i. The pair formed by the structure (graph) and the collection of local distributions, $P(x_i \mid \Pi_i)$, for each node in the domain, constitutes the Bayesian Network for that domain. Figure 1 plots a sample Bayesian Network where variables **B** and **D** are independent, variable **C** is only dependent of **D**, and variable **A** is conditionally independent of the variable **D** given **B** and **C** variables.

Exploring conditional independence between variables, this formalism provides a compact description of the joint distribution, explicitly representing local dependence; this characteristic is explored in the design of efficient inference algorithms.

In the context of ITSs, Bayesian Networks have been applied to user modelling (VanLehn *et al.* 1998) in a diagnostic perspective: given a student action (symptom) the network provides the most likely state of knowledge (diagnosis). In our work, BNs are used to model the user, the structure of knowledge and pedagogical options. The values related to student status are taken from the user interaction with the tutor, further described in section 3.3.

3 THE DESIGN MODEL

Our design model presents a unified solution for the creation of ITSs based on Bayesian Networks.

This model is developed taking into account two phases of the ITS life cycle: the *design* phase of ITSs, where we focus on the modularity and rapid development of systems; the *usage* phase as an auxiliary tool, directed towards the adaptability to the user.

A student will use an ITS to receive a specialized learning orientation, on a given domain, tuned to the user unique learning style and state of knowledge.

We envisioned the design of an ITS system by decomposing it in three conceptual modules: one associated to the user; another concerning the contents related to the knowledge representation; and finally, an inference mechanism which form the core of the system, linking the other modules. This conceptual design will be further explored in a modular architecture implemented as a *shell*.

3.1 USER MODELLING

To model the user learning style and state of knowledge we define a set of variables both generic (associated with the learning style) and content specific (associated with the state of knowledge). These variables will be part of a BN (see section 3.3) to infer conclusions about the student.

Generic user related variables (learning style variables) include:

- mean time per page—indicates the average time a student spends on a page.
- superficial scan—identifies a student preference in passing the contents in a shallow way.
- answers—indicates if a user answers to questions.

For each topic exists the list of variables include (state of knowledge variables):

- spent time—specifies the time spent by the user reading/interacting with the specific topic.
- topic deepness level—refers to the tendency of a user to deepen a topic.
- wrong answers—number of questions not answered correctly.
- right answers—number of correct answers.

These variables keep information about the user interaction with the ITS and are used to adapt the presentation of contents to the user profile.

3.2 KNOWLEDGE DESIGN

The repository of knowledge stored in the *knowledge base* is structured as: (a) a set of *topics* that represent atomic content parts (b) *knowledge map* that describes the interdependence between *topics*. Each *topic* is composed by: (a) a hypertext page including text, pictures and other multimedia items; (b) a set of questions.

Steps towards the construction of the *knowledge base* are as follows: *knowledge map* definition; topic creation; design of individual topic text, pictures and multimedia items.

3.3 BAYESIAN INFERENCE

Our model of ITS is completed with the definition of a BN that relates the *knowledge map* with *topic* related variables and integrate the learning style of the user with the pedagogical options.

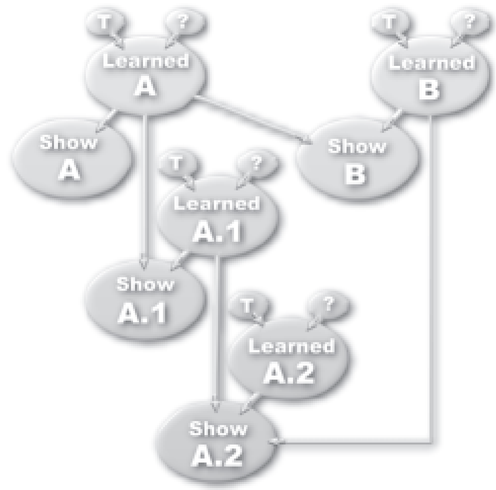

Figure 2: Bayesian Network from a sample knowledge base.

Consider an ITS with the following topics:

- Topic A
  - Subtopic A.1
  - Subtopic A.2
- Topic B

In this example, a dependence relation exists between Topic B and Topic A.2, meaning that in order to entirely understand topic A.2 it is necessary to learn topic B.

A simplified Bayesian Network for the previous example is shown in figure 2. The **learned** nodes represent the degree of belief that a certain topic has been learned. These belief values are dependent on the time spent on the corresponding topic (**T** in figure 2) and on the answered questions (**?** in figure 2).

We define each **T** and **?** in equations 2 and 3 respectively.

$$\mathbf{T} = \frac{Time\ in\ topic}{Standard\ time\ in\ topic} \tag{2}$$

$$\mathbf{?} = \frac{Right\ answers - Wrong\ answers}{Number\ of\ question} \tag{3}$$

Each **show** node specifies the degree of belief in that the related topic should be learned. These nodes depend on its **learned** parent nodes and on the **learned** node of the same topic.

Figure 3: Bayesian network for the pedagogical actions.

The student learning style is inferred through a set of BNs using user related variables. Figure 3 shows a subset of this BN illustration *deepen* topic and *review* topics.

4 THE INTELLIGENT TUTORING SYSTEM ARCHITECTURE

According to the previous design model the proposed ITS architecture adopts a modular structure composed by five basic blocks (figure 4), namely: (a) **User Model**; (b) **Knowledge Base**; (c) **Adaptation Module**; (d) **Pedagogical Module**; (e) **Presentation Module**. The internal structure of these modules is independent of the tutor specific content, thus equal for all the ITS implemented using this approach, constituting a *shell*. Implementation aspects of this *shell* are detailed in section 5.

The **User Model** is the part of the ITS where the variables defined in section 3.1 are kept. This module retains user information from the interaction with the tutor.

The content subjects to be taught are kept in the **Knowledge Base** structured as a *knowledge map* and a set of *topics*. This is the unique module in a working ITS that is dependent of the contents, producing the desired separation between the domain application dependent part and the generic part, the *shell*.

User assessment is performed in the **Adaptation Module** which combines information from the Knowledge Base with user data (from the user model) in a BN, inferring the user state of knowledge.

The **Pedagogical Module**, also supported in BNs, uses the data provided by the three last modules and creates a set of suggestions of what the student should do next.

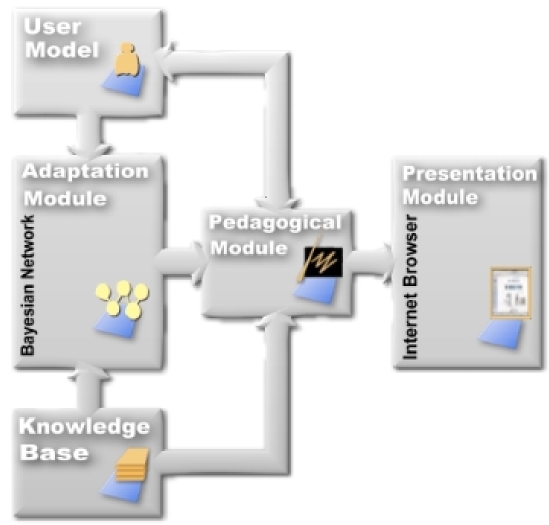

Figure 4: Intelligent tutoring system architecture.

The result of the ITS reasoning process is the page composed and shown in the **Presentation Module**. It collects the topic and respective questions to present, the pedagogical suggestions and the most likely step to be taken.

This architecture serves to isolate the different parts of the system defining their functionalities and specifying the implementation directions.

5 THE ITS SHELL IMPLEMENTATION

The proposed architecture is implemented in *Java* and *JavaScript* being accessible through an Internet browser. BNs structure and their respective inference algorithms are built using the *JavaBayes* (Cozman 1998) (Dechter 1996) program also implemented in *Java*.

- User model

  The user model is kept in a *Java* data structure that has a set of general variables and a set of variables related to the topics from the knowledge base.

- The Knowledge base

  The internal structure of the knowledge base is developed under an object oriented approach, and for each content part that is present in the ITS, exists an object associated with an html page, a set of questions and variables.

  The structure of this `Page` object is as follows:

```
Page{
  Title
  Html File

  Questions List
  Number Of Questions
  Wrong Answers
  Right Answers

  Time Spent In Page
  Visited Page?
  Standard Page Time

  Parent Pages List
  Number Of Parents
}
```

Where `Questions List` is a list of objects of the type `Question` with the following items:

```
Question{
  Question Text
  Help In Case Of Error
  Correct Answer
  Used Question?
}
```

The *html* page is composed in a separate program and the questions are designed to be answered with yes or no choices. Information in the `Page` structure is passed to the other modules of the system.

At each user action these variables are updated in order to have this information available for the other modules.

- The Adaptation Module

  From the structure of the knowledge base, a BN is automatically built to infer if each topic is learned and infer the next topic to be learned.

- Pedagogical module

  The role of the pedagogical module is to choose between one action over the current topic and the decision of showing a new topic: (a) deepen a topic; (b) shallow scan; (c) review previous topic; (d) show a new topic.

  These options are defined using the BNs in figure 3; the propagated probabilities of possible actions are compared with the probabilities of the **show** nodes in the adaptation model. The highest probability node is selected using equation 4.

$$action = arg \max \left\{ \begin{array}{l} P(\mathbf{deepen\ topic}) \\ P(\mathbf{shallow\ scan}) \\ P(\mathbf{review}) \\ \max_i [P(\mathbf{show_i})] \end{array} \right\} \quad (4)$$

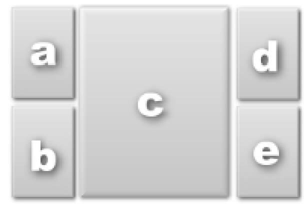

Figure 5: Presentation module parts: (a) navigation; (b) questions; (c) topics (d) suggestions (e) index.

  If the decision is **deepen topic** the tutor presents the descendent page of the actual page. In case of **shallow scan**, the next page in the index is presented. The **review** decision selects the parent page most likely to be shown (the parent page with the highest **show** node probability).

  With this structure the ITS enables a guided navigation through the topics, and adapts the path to the student learning style.

- The Presentation Module

  The user interacts with the ITS through the presentation module, the system part that composes the information to be shown, using the information generated by the previous modules. The presentation module enables the user to accept the guidance of the ITS or directly select a topic.

  The interface consists of a screen divided into five parts (figure 5): (a) *navigation*; (b) *questions*; (c) *topics*; (d) *suggestions*; (e) *index*.

  The *navigation* part has three buttons: the *next* button leads to the presentation of the next page according to the pedagogical direction (formula 4); the other buttons follows to the *last* subject accessed or jumps to a parent subject so that the user can *review* related subjects.

  The area of the screen dedicated to *questions* is dynamically changed, drawing user attention. These questions are optional, but if answered they produce information that enables the tutor to better guide the user.

  The *topic* is displayed in an *html* page positioned at the center of the browser.

  The *suggestions* displayed are the three higher probability decisions taken from the pedagogical module.

  In the *index*, the ITS presents a conventional structure, like in a book, that enables the user to freely navigate through the topics. Every user action is sent to the user model, so that the variables related to the user are updated.

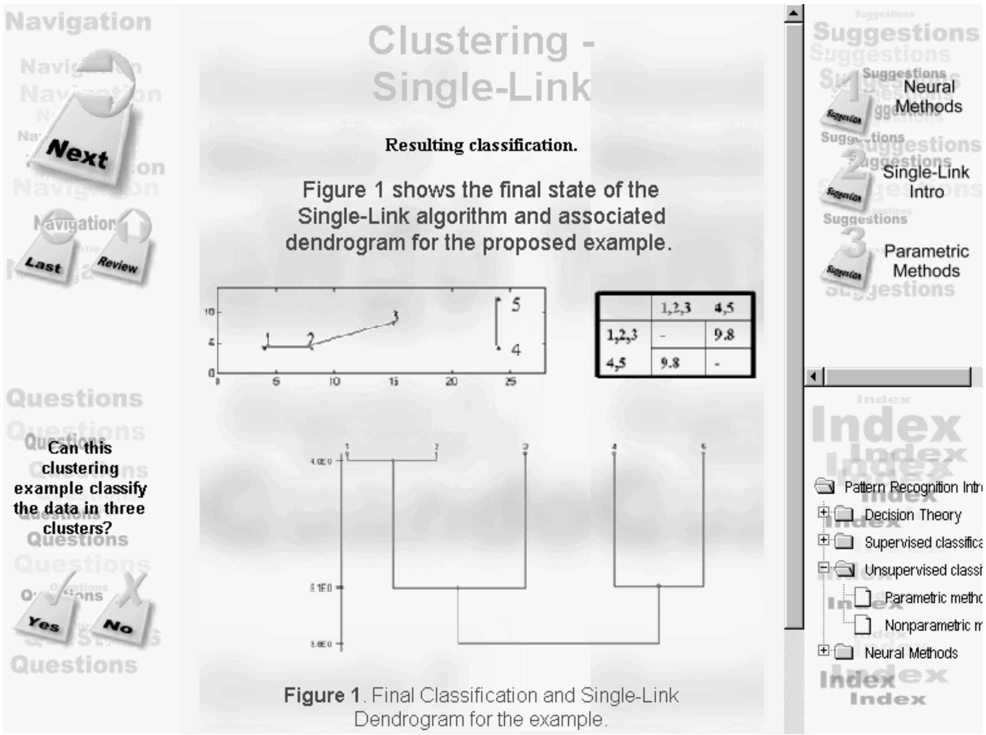

Figure 6: Interface of the illustrative example.

6 ILLUSTRATIVE EXAMPLE

In this section we show an example of an ITS, developed using the previously described *shell*, in the area of *Pattern Recognition*.

In the design phase the contents were produced using available tools to create *html* pages and graphics. We added a set of questions for each topic, and the interdependence among the different topics were identified and mapped into the tutor.

After the design phase the system was used as an auxiliary tool. Figure 6 shows a typical screen of the system in the topic related to *clustering algorithms*.

When the user starts using the tutor, he/she is asked to do a short test to assess his prior knowledge in the subjects to be taught. This test will initialize variables in the BN. The ITS then starts showing the topics, making question and suggesting paths. It acquires information about the user via the interaction with the browser in the presentation module. At the end of the tutor usage, an evaluation test will compare the first results with those of the final test.

7 CONCLUSIONS

In this paper we proposed a modular architecture for the design of ITS, under a Bayesian approach. Adopting Bayesian Networks as the basic formal device for modelling and reasoning under uncertainty, two main goals were pursued: personalization and adaptation of the system to student profiles and state of knowledge; facilitate the design process of ITS.

Contributions towards these objectives included:

- definition of a global ITS model resulting from the integration of specialized models, namely: the user model, the knowledge base and the Bayesian inference.
- providing these models with learning capabilities by means of probabilistic inference based on Bayesian Networks.
- mapping of this conceptual model into a modular architecture clearly separating domain application blocks from generic knowledge representation and inference mechanisms common to all applications, implemented as a *shell*.

The implementation of the *shell* using *Java*, extends the usage of the ITS as an *e-learning* tool.

Structuring knowledge as atomic knowledge pieces hierarchically organized according to precedence rules in an interdependence *knowledge map*, *html* files were chosen for the design of content pages enabling the incorporation of a versatile set of multimedia publishing resources.

An example of the application of the proposed methodology and *shell* in the domain of Pattern Recognition was shown illustrating the design and final usage phases of an ITS.

Ongoing work includes refinement of the present model, and special attention is given to the adaptability of the model to produce internet programs to be introduced in existing *e-learning* tools.

REFERENCES

Beck, J.; Stern, M.; and Haugsjaa, E. 1996. Applications of AI in Education. In *Proceedings of the Third International Conference on Intelligent Tutoring Systems*.

Bennett, F. 1997. *Computers as Tutors: Solving the Crisis in Education*. First Monday. http://www.firstmonday.dk/issues /issue6/intro/index.html.

Brusilovsky, P.; Schwarz, E.; and Weber, G. 1996. Elm-art: an intelligent tutoring system on world wide web. In C. Frasson, G. Gutier, A. L., ed., *Intelligent Tutoring Systems, ITS'96*, pp. 261–269.

Buchanan, B. G., and Shortliffe, E. H. 1985. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

Burns, H. L., and Capps, C. G. 1988. Foundations of Intelligent Tutoring Systems. In Polson, M. C., and Richardson, J. J., eds., *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates. chapter 3, 55–78.

Cozman, F. 1998. Javabayes version 0.34, bayesian networks in java usermanual. http://www.cs.cmu.edu/ ˜fgcozman/home.html.

Dechter, R. 1996. Bucket elimination: a unifying framework for several probabilistic inference algoritms. In *Twelfth Annual Conference on Uncertainity in Artificial Inteligence*.

Fleming, J. L., and Horwitz, C. 1996. Applications of the Rapid Intelligent Tutoring System Development Shell (RIDES). In *ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs*.

Frölich, P., and Nejdl, W. 1997. A Database-Oriented Approach to the Design of Educational Hyperbooks. In *Proceedings of the workshop: Intelligent Educational Systems on the World Wib Web, 8^{th} World Conference of the AIED Soceiety*.

Fred, A. 1994. Expert systems: Conceptual, methodological and management aspects. Technical Report Project HC 1014, European Neurological Network.

Frey, B. 1998. *Graphical Models for Machine Learning and Digital Communication*. MIT Press.

Jensen, F. 1997. *An Introduction to Bayesian Networks*. Springer.

Murray, T. 1999. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education* 10: pp. 98–129.

Nilson, N. 1998. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann.

Pearl, J. 1998. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Russel, S., and Norvig, P. 1995. *Artificial Intelligence: a modern approach*. Prentice Hall.

Schulze, K. G.; Shelby, R. N.; Treacy, D. J.; Wintersgill, M. C.; Vanlehn, K.; and Gertner, A. 2000. Andes: An intelligent tutor for classical physics. *The Journal of Electronic Publishing, University of Michigan Press*. http://www.press.umich.edu/ jep/06-01/schulze.html.

Ubell, R. 2000. Engineers turn to e-learning. *IEEE Spectrum* 37(10): pp. 59–63.

Urban-Lurain, M. 1996. Intelligent tutoring systems, an historic review in the context of the develpment of artificial inteligence and educational psychology. Technical report, Department of Computer Science and Engineering, Michigan State University. http://aral.cse.msu.edu/ Publications/ITS/its.htm.

VanLehn, K.; Niu, Z.; Siler, S.; and Gertner, A. 1998. Student modeling from convencional test data: A bayesian approach without priors. In Goettl, B. P.; Halff, H. M.; edfield, C. L.; and Shute, V. J., eds., *Intelligent Tutoring Systems, 4^{th} International Conference, ITS'98*, volume 1452. Springer.

VanLehn, K. 1988. Student Modeling. In Polson, M. C., and Richardson, J. J., eds., *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates. chapter 3, 55–78.

Zimmerman, H. J. 1991. *Fuzzy Set Theory*. Kluwer, 2nd edition.