# Intelligent Guide: Combining User Knowledge Assessment With Pedagogical Guidance

Ramzan Khuwaja[1], Michel Desmarais[2], Richard Cheng[2]
[1]Cognologic Software Inc.,
3620 Lorne Crescent, #216, Montreal, Qc, H2X 2B1
Phone: (514) 842-1527
Email: khuwaja@steve.iit.edu

[2]Computer Research Institute of Montreal,
1801 McGill College Avenue, #800, Montreal, Qc, H3A 2N4
Phone: (514) 398-1234; Fax: (514) 398-1244
Email: desmar@crim.ca, rcheng@crim.ca

**Abstract.** Despite their many successes, Intelligent Tutoring Systems (ITS) are not yet practical enough to be employed in the real world educational/training environments. We argue that this undesirable scenario can be changed by focusing on developing an ITS development methodology that transforms current ITS research to consider practical issues that are part of the main causes of underemployment of ITSs. Here we describe an ambitious research project to develop an ITS that has recently completed its first phase of development at the Computer Research Institute of Montreal. This project aims to address issues, such as, making ITS handle multiple domains, developing cost-effective knowledge assessment methodologies, organizing and structuring domains around curriculum views and addressing the needs of users by considering their immediate goals and educational/training settings. This paper concentrates on the outcomes of the first phase of our project that includes the architecture and functionality (specially user knowledge assessment and pedagogical guidance) of the Intelligent Guide.

## 1 Introduction

The field of Intelligent Tutoring System (ITS) is almost three decades old but only a handful of ITSs built so far made their way into real educational environment [12; 7]. ITS community is beginning to aware of this concern and a number of researchers have already started to suggest different way outs for this situation (see for example, [7]). The field of ITS has great promise and potential to be effective in both educational and training worlds but these two worlds have their own unique demands and requirements. One important step forward for ITSs, to be more practical, can be achieved by identifying and addressing issues that are common to these two worlds.

One of the important issue that the field of ITS is currently facing is the "system problem," [12], i.e., "the design of ITS needs to be done with consideration for how it will be used within an educational [or training] system, rather than just developing it as a stand-alone entity" (p. 53).

The second issue is of making ITS capable of handling multiple domains at the same time. Most of ITSs developed so far deal with only single domain of knowledge. On the contrary, many potential educational and training applications demand knowledge in multiple domains of expertise.

Only cost-effective computer-based systems are capable of justifying their acceptance in real world situations. This is specially true for ITSs that make their way into training domains. These systems are still expensive to build, mainly as a consequence that the development of an ITS still starts from scratch. There is no ready-to-use methodology and technology available that a developer can use to start with.

Two other issues that are very closely related to the “system problem” are the effectiveness of an ITS and its acceptability by the user. An effective ITS should produce a high gain in the user’s knowledge/skills. This can only happen if the system provides enough motivation for the user and justify its usage in the pursuit of the user’s goal(s).

This paper describes an ambitious research project to develop an ITS that has recently completed its first phase of development at the Computer Research Institute of Montreal (CRIM). This project addresses some of the important issues discussed above.

## 2 Background

A solution to issues, described above, lies on the development of a generic methodology to design an ITS. This methodology should have power to handle knowledge from multiple domains. Many ITSs developed so far concentrated on a single domain. In general, this resulted in a powerful model of expertise but the methodology used to develop that model becomes restricted to that specific domain. A flexible and generic methodology can result in cost-effective systems for real world educational/training environments. Developing one such generic methodology require detailed understanding of the functioning and architecture of ITSs.

ITSs are complex systems. A common trend in the ITS community is to organize the development of a tutoring system around four functional components [13]: diagnostic, (domain) expert, pedagogy, and communication modules. In order for an ITS development methodology to be flexible and generic it needs to prescribe the development of each of these modules. The nature of these modules depends upon the consideration for the overall architecture for an ITS. Wenger in [13] has characterized ITSs as consisting of either model-based or curriculum-based architectures. A model-based ITS emphasizes the model view of the domain expertise. Some example ITSs in this class are: Lisp Tutor [1], QUEST [14], CIRCSIM-Tutor [8]. The curriculum-based ITSs, on the other hand, emphasize the curriculum view of the domain expertise, example ITSs in this class are: BIP [2], WUSOR [6], MHO [9]. In Wenger’s [13] term the curriculum-based architectures “emphasize the notion of *lesson* rather than that of *model* as a reservoir of domain knowledge (p. 149). We believe that the knowledge domains required for many real world educational/training applications land well for curriculum-based ITSs. Another advantage of this approach is that, due to its very nature, it emphasizes the body of knowledge that specifies the goal structure for the system [9]. As noted by Lesgold in [9] a vast majority of courses developed in the educational world use explicit curriculum. A use of explicit curriculum in an ITS, it is hoped, will facilitate its integration in an educational setting.

### 2.1 Intelligent Guide: An ITS that facilitates acquiring mastery level competence in a domain

This paper describes a computer-based educational system, Intelligent Guide (IG), that is currently under development at the Computer Research Institute of Montreal (CRIM). The first phase of development for this system has recently completed. This paper concentrates on the outcomes of the initial phase of this project.

The long term goal of Intelligent Guide is to develop a generic Intelligent Tutoring System (ITS) that could provide user knowledge assessment and pedagogy guidance for a number of domains that require the user to master a number of concepts or skills to achieve a satisfactory level of competence in the domain. Our research has this definite objective of bringing this system into real world educational/training environment.

Intelligent Guide has a generic curriculum-based architecture. It is designed to operate with a general knowledge assessment method [3]. This method uses a kind of overlay type user model. One of the advantage of this method is that it has power to be effective in multiple domains. We are currently experimenting the Intelligent Guide with a specific domain, the Graduate Management Admissions Test (GMAT).

# 3 Architecture of Intelligent Guide

The typical knowledge domains we are envisioning for the Intelligent Guide will consists of a large body of concepts and/or skills that a student needs to master. Learning of a concept /skill will be tested by asking one or more questions to the student. All the concepts and skills can be arranged in a network of nodes that are connected by several relationships (e.g., part of, prerequisite, analogy, co-topic). In other words, knowledge in our domains can be represented by a type of curriculum structure.

The major objective of Intelligent Guide is to provide guidance to the user in the pursuit of achieving satisfactory level of competence in a domain. Intelligent Guide is like a tutoring assistance that assesses the knowledge state of the student for a domain. Based upon this assessment it points out areas in the knowledge domain that require attention from the user. Unlike the currently available software products (see Sec. 2.1), the degree of attention required by the user for these areas is part of the feedback provided by Intelligent Guide. Further, depending upon the user's choice this tutoring assistance can invoke a tutorial session for a domain concept/skill that needs to be learned/mastered by the user. Intelligent Guide is not designed to provide a full delivery of contents for each domain topic but rather a brief but comprehensive overview of major concepts required. We assume that the users in our case know the basics of the knowledge domains. These users are mainly looking for assessment of their knowledge level and an individualized (active) review of the different domains. We do not attempt to restrict the user to learn domain contents from only Intelligent Guide, in fact this system provides pointers to commonly available books and other forms of resources for the user to acquire advance knowledge of the domain.

An ultimate goal of Intelligent Guide is to encourage the user to periodically use this system while participating in a preparatory course or preparing for a test like GMAT. In this way the user would have an opportunity to keep track of his/her progress in learning the domain material. Considering this goal of our system it is imperative to continuously evaluate the knowledge state of the student to individualize feedback/guidance.

Fig. 1 shows a scenario for Intelligent Guide. In this scenario the Intelligent Guide is like a shell, missing the domain knowledge. In order for this system to be functional in a domain, the knowledge for that domain needs to be fed into it. We assume that this knowledge comes from a course designer in that domain. Currently we are developing a software tool, course development environment, that allows a course designer to structure and organize his/her view of the domain knowledge. This tool then automatically transforms this knowledge into an intermediate form recognizable by Intelligent Guide. When Intelligent Guide is invoked by the user, it reads this intermediate form of domain knowledge to create a knowledge base which is used to perform reasoning in the domain.

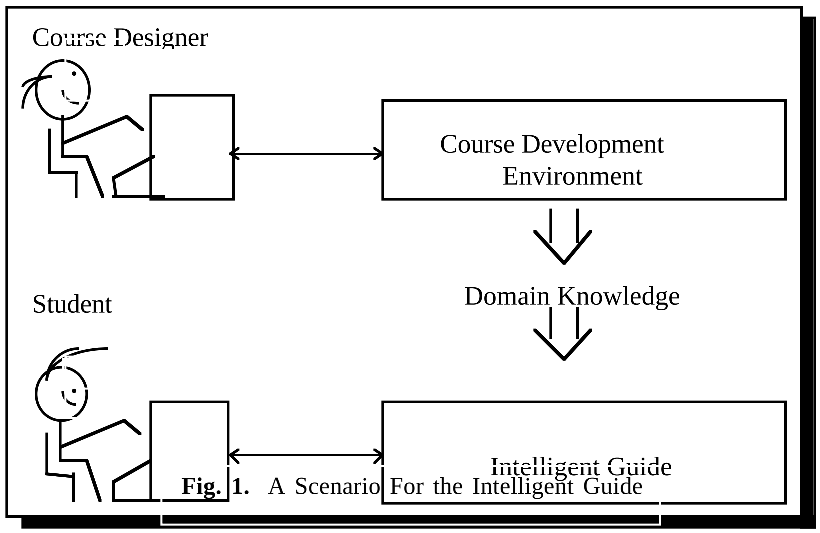


**Fig. 1.** A Scenario For the Intelligent Guide

Fig. 2 shows the internal architecture of Intelligent Guide. The arrows in this figure represent the data flow paths between components of this system. As it is shown in this figure, this system consists of six major components. Intelligent Guide is designed using an object-oriented methodology and is implemented using the C++ programming language.

The architecture of Intelligent Guide is influenced by Lesgols's research (see [9]). Besides this formalism two very practical issues influenced this architecture, these are: making Intelligent Guide available on multiple platforms and plugging it with multiple user interfaces. Currently Intelligent Guide runs on UNIX and Windows environments and uses a Web-based user interface.

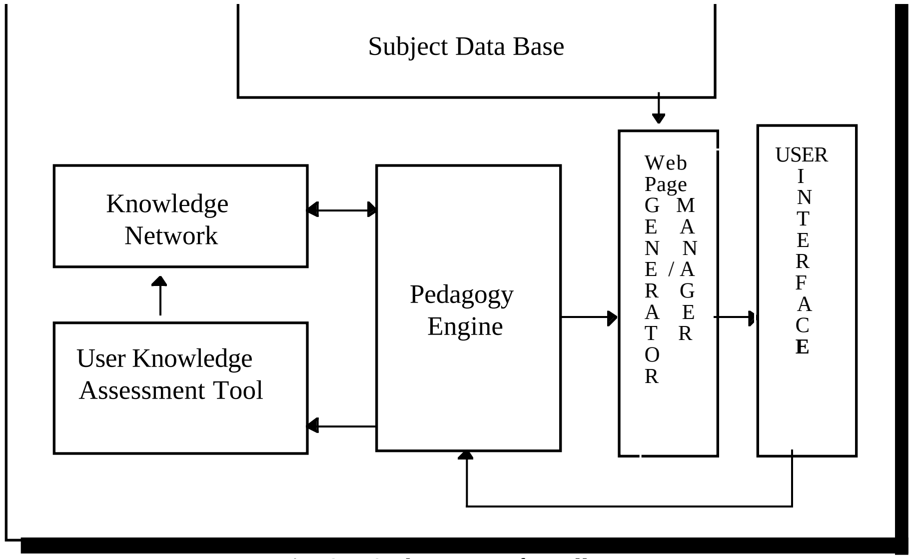


**Fig. 2.** Architecture of Intelligent

The knowledge network of Fig. 2 contains Knowledge Units (KU) or domain entities (e.g., domain concepts) that are connected via multiple domain relationships (e.g., prerequisite). This network is created from an intermediate form of domain knowledge provided by the curriculum development environment (see Fig. 1). This knowledge network is basis for the domain intelligence in the system.

The subject data base is the second component that uses output from the course development environment. This data base consists of actual content of domain entities that is communicated to the user, for example, content for various GMAT questions is kept in various files of this data base. When the pedagogy engine selects a question that needs to be asked to the user, an appropriate file(s) containing this question is accessed from the subject data base.

The user knowledge assessment tool is responsible for assessing the knowledge state of the user. It uses a specific technique to keep track of the changing state of the user's knowledge. Sec. 4 describes in detail the theoretical background and working of this component.

The pedagogy engine is the heart of Intelligent Guide. It is responsible for interpreting the knowledge state of the student and, depending upon the currently selected goals and the state of interaction with the user, it determines "what to do next" in a session. Within the architecture of Intelligent Guide this component is also responsible for coordinating the communication between other components of the system. Sec. 5 describes the theoretical background and functioning of this component in detail.

Once the pedagogy engine has decided about what and how to present contents to the user, it sends messages to the web page generator and manager (see Fig. 2) to compose a HTML (Hyper Text Markup Language) page for the user. It is this component that queries the subject data base to develop a HTML page. Once a page is generated, it is sent to the user interface (see Fig. 2), the component that handles communication with a HTML server.

## 4 Knowledge Assessment in the Intelligent Guide: A Generic User Model

One of the fundamental component of the Intelligent Guide is the user knowledge assessment module (see [3; 4]). This module is responsible for providing a user profile of the knowledge network's state of mastery. Based upon the information it receives from the pedagogy engine about what knowledge unit (KU) is mastered by the user, the knowledge assessment module will infer the likelihood that every other KU is mastered.

The knowledge assessment module adopts the overlay approach to defining the whole domain knowledge. It uses a view of the knowledge network that organizes fine-grained Knowledge Units (KU), or nodes, into a *knowledge structure* [5] that represents the order in which KU are learned. An individual's knowledge about the domain, i.e., a knowledge state, is modeled by a collection of numerical attribute values attached to the nodes. Each value indicates the likelihood (i.e., probability) of a user's knowing a specific KU. In the knowledge structure, KU are connected by implication (precedence) relations. An implication relation is, in fact, a gradation constraint which expresses whether a certain concept has to be understood before another difficult one, or whether a certain skill is acquired prior to an advanced one. It is these implication relations that enables the inferences about the mastery of KU.

In contrast to other work that also adopted similar approaches to knowledge assessment (see in particular [6]), the knowledge structure is induced entirely from empirical data composed of samples of knowledge states. Because the knowledge structure induction process is automatic, it allows a much larger number of KUs to be included than other approaches which "manually" build the structure with the help of domain experts. A more detailed treatment of the knowledge structure construction method is given in [3].

Once a knowledge structure is obtained, it can be used as a basis for knowledge assessment. The knowledge state of a user is built and updated as soon as some observations are made (e.g., questions and exercises are answered). Each observation can be viewed as a piece of evidence. This new information may be propagated to other nodes in compliance with the gradation constraints (inference structure). Standard evidence propagation techniques can be used to perform this process (a number of such techniques can be found in [11]). Again the reader is referred to [3] for further details.

## 5 Pedagogy Guidance

This section describes the design and functioning of the pedagogy engine (see Fig. 2). The major function of this component is to provide pedagogy guidance to the user. Its behavior depends upon (1) the domain knowledge in the knowledge network, (2) user's actions, (3) an assessment of the user's knowledge state, and (4) the pedagogy knowledge represented as rules in this component.

Like Lesgold (see [9]) we also make distinction between the domain knowledge and curriculum (or goal structure) in our system. The domain knowledge provides the domain intelligence that the pedagogy engine tries to impart to the user but it is the curriculum knowledge that provides a mean of planning a session and making moment-to-moment decisions while communicating with the user.

Before we further describe the pedagogy engine, let us briefly sketch the organization of the knowledge network. This network consists of domain entities (Knowledge Units - KU - or nodes) and relationships between these entities. There are two major sections of this network, we call them, subject-matter hierarchy and question hierarchy. The subject-matter hierarchy is organized into four layers: course layer, section layer, topic layer, and concept layer. The question hierarchy consists of nodes for questions, solutions to questions, and problem-solving strategies to solve a question. Questions in this hierarchy test knowledge for various domain entities in the subject-matter hierarchy. These two hierarchies intersect at nodes (e.g., real number division) representing the domain entities that need to be tested to evaluate the user's competence in the domain.

This type of break down of subject-matter contents is very important from our point of view because (1) this is a common way of organizing domain knowledge by the instructional system designers, (2) it also provides a view of curriculum that the system needs to consider before engaging the user in a session, (3) it provides a vehicle for generating system goals for the Intelligent Guide.

It is one of the important goal of our research that the Intelligent Guide be designed quickly and efficiently for a domain by minimally restructuring the contents of a book to develop a domain knowledge network for our system. We believe that the domain knowledge organization used in our system provides a generic framework that can be used to transform pre-organized knowledge in a book into a knowledge network for the Intelligent Guide. Using this framework we have transformed the math (review) section of the book *The Official Guide for GMAT Review* by the Graduate Management Admission Council to develop the current network for the Intelligent Guide.

The pedagogy engine is a hierarchical, incremental instructional planner [10]. It uses the subject matter classification of the knowledge network to develop plan at four different levels: course, section, topic, and concept (notice the similarity of these levels with the layers in the knowledge network). Because of this multiple-level organization it is possible for the pedagogy engine to develop plan at the global level to consider the overall objective of a session and at the interaction level where moment-to-moment decisions are made to continue a form of dialogue with the user.

The decision making process within the pedagogy engine is fueled by the pedagogy rules. These rules reside in a rule base. The interpretation and execution of these rules is carried by a rule base engine. We have classified these rules as: meta-rule, session management rule, goal refinement rule, goal execution rule, and the student initiative handling rule. Most of these categories are self explanatory. Meta-rules are rules that decide which rule category to consider in view of the current state of the system. The student initiative handling rules decide how to respond to the user's actions (e.g., questions, requests). An English version of an example goal refinement rule is as follows.

*IF current goal is to guide-the-student, and*
*current planning level is topic-level*
*THEN create sub-goals:*
* * sequence currently focused topics*
* * select a focused topic, and*
* * present (to the student) the selected topic*

The behavior of the pedagogy engine can also be viewed as setting and execution of goals at four different decision making levels. This multi-level planning model is based on the decision making process commonly used by the instructional designers.

Intelligent Guide provides option for the user to set some of the goals at each level. This means that the user can select a course, a section, a topic, and/or a concept to pursue with. This, we hope, will provide enough motivation to the user to take change of the system to achieve his/her goals.

The organization of the pedagogy engine is very generic (i.e., domain independent). We believe that Intelligent Guide can handle most of the domains whose contents can be organized into a curriculum-type structure described above.

# 6 Conclusion

One of the major objective of our current research and development track is to bridge the gap between research and practical application of intelligent computer-based teaching/training systems. We are striving to develop a methodology and technology for ITS that could facilitate to develop effective and economical systems that could be easily integrated in a real world teaching/training environments. Unlike the current trend in the ITS community we are aiming for techniques that may not be very sophisticated but effective enough to achieve our goals. This require us to adopt and transform currently available ITS research and also to develop new techniques for our purposes.

Most of ITS developed so far were oriented towards developing methodologies for single domains. In order to achieve our goals we believe that we need to take the next logical step towards ITS research, i.e., to orient our methodology towards a group of domains and educational/training situations. Our group oriented methodology, we hope, will be a good candidate to effectively address the issues raised in the introduction (see Sec.1).

We have selected knowledge/skill evaluation tests like GMAT and preparatory courses as our target domains for our research and development. Moreover, we are currently focusing on knowledge evaluation and pedagogical guidance as the main functions for our system. Considering our target educational situation we have observed that our domains could be organized to explicit their curriculum structures. These structures provide a very fertile ground for our knowledge assessment methodology which already, theoretically and practically, have been proved to be generic and effective (see for example [3]). We have adopted current ITS research for developing curriculum-based ITS to provide pedagogical guidance for Intelligent Guide.

Our journey towards our goals has just started. In our first phase of development we have started to address some of our theoretical and practical issues. In our second phase we are aiming to develop a set of tools that could help course designers to develop contents for our system. We are also aiming to test our system in a variety of educational settings that are originally intended for our project. We do realize that we are aiming for an ambitious project but with our clear, cautious, and practical approach we hope to succeed in our goals.

## References

1. Anderson, J. R., Boyle, C. F., Corbett, A. T. & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. In Clancey, W. J. & Soloway, E. (Eds.). *Artificial intelligence and learning environment* (pp. 7-49). Cambridge, MA: The MIT Press.

2. Barr, A., Beard, M. & Alkinson, R. C. (1976). The computer as a tutorial laboratory: The Stanford BIP project. *International Journal of Man-Machine Studies*, 8, 567-596.
3. Desmarais, M. C., Maluf, A. & Liu, J. (to appear). User-expertise modeling with empirically derived probabilistic implication networks. *User-Modeling and User Adaptive Interaction*.
4. Desmarais, M., Giroux, L., Larochelle, S. & Leclerc, S. (1988). Assessing the structure of knowledge in a procedural domain. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society* (pp. 475-481).
5. Falmagne, J., Doignon, J., Koppen, M., Villano, M. & Johannesen, L. (1990). Introduction to knowledge spaces: how to build, test and search them. *Psychological Review*, 97, 201-224.
6. Goldstein, I. P. (1982). The genetic graph: A representation for the evolution of procedural knowledge. In Sleeman, D. & Brown, J. S. (Eds.), *Intelligent tutoring systems* (pp. 51-77). London: Academic Press, Inc.
7. Jones M. (1992). Instructional systems need instructional theory: Comments on a truism. In Scanlon, E & O'Shea, T. (Eds.), *New directions in educational technology* (pp. 1-13). Berlin, Germany: Springer-Verlag.
8. Khuwaja, R. A. (1994). *A model of tutoring: Facilitating knowledge integration using multiple models of the domain*. Ph.D. Thesis, Computer Science Department, Illinois Institute of Technology, Chicago, Illinois.
9. Lesgold, A. (1988). Towards a theory of curriculum for use in designing intelligent instructional systems. In Mandl, H. & Lesgold, A. (Eds.), *Learning issues for intelligent tutoring systems* (pp. 114-137). New York: Springer-Verlag.
10. Murray, W. R. (1988). *Control for intelligent tutoring systems: A comparison of blackboard architecture and discourse management networks*. Technical Report R-6267, FMC corporation.
11. Pearl , J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.
12. Reigeluth, C.M. (1992). New directions for educational technology. In Scanlon, E & O'Shea, T. (Eds.), *New directions in educational technology* (pp. 51-59). Berlin, Germany: Springer-Verlag.
13. Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Los Altos, CA: Morgan Kaufman.
14. White, B. Y. & Frederiksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. In Clancey, W. J. & Soloway, E. (Eds.). *Artificial intelligence and learning environment* (pp. 7-49). Cambridge, MA: The MIT Press.