

## Association for Information Systems AIS Electronic Library (AISeL)

---

AMCIS 1998 Proceedings

Americas Conference on Information Systems  
(AMCIS)

---

December 1998

# A Prototype Intelligent Tutoring System for Programming Languages: Adding a User Model and Feedback to Instructional Transaction Shells

Jens Liegle  
*Kent State University*

Gregory Madey  
*Kent State University*

Follow this and additional works at: <http://aisel.aisnet.org/amcis1998>

---

### Recommended Citation

Liegle, Jens and Madey, Gregory, "A Prototype Intelligent Tutoring System for Programming Languages: Adding a User Model and Feedback to Instructional Transaction Shells" (1998). *AMCIS 1998 Proceedings*. 362.  
<http://aisel.aisnet.org/amcis1998/362>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1998 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# **A Prototype Intelligent Tutoring System for Programming Languages: Adding a User Model and Feedback to Instructional Transaction Shells**

**Jens O. Liegle**  
**Gregory R. Madey**  
Kent State University

## **Abstract**

*Intelligent tutoring systems and knowledge management systems are becoming increasingly important, especially their connection to training-on-demand systems. New developments like the Internet and advances in expert systems allow the deployment of intelligent interfaces, which can present the knowledge accumulated in a database. This paper proposes a prototype that utilizes these new technologies to extend Merrill's work (1992) on instructional transaction shells. These shells embody instructional techniques that are independent of the knowledge they are teaching, and thus reusable. The prototype includes a self-learning expert system for user modeling, thus allowing for a personalized user interface and presentation of content.*

## **Introduction**

A large number of different disciplines are investigating the area of knowledge management and intelligent tutoring systems (ITS). Education, library and information science, computer science, and management information systems are just a few of them. The goal is not only to have a large database of information and facts, but also to provide links between the different pieces. If combined with a computer based training system, these knowledge systems can be used for training on demand, a very promising application domain. Studies show that ITS can support learning, e.g. Fletcher-Flinn and Gravatt (1995) reported that in general there is a positive learning effect of computer assisted learning. However, the vast majority of such systems in industry and commerce are not as successful as hoped (Martinsons and Schindler, 1995). Still, the online training industry is expected to grow to \$28 billion per year worldwide by the year 2001 (Dillon, 1997). Different, but similar, estimates are reported by Herther (1997) who shows predictions that \$12 billion will be spent on online business training in 1998, and that half of all corporate training will be online by the end of the century. She also reports that all Fortune 1000 companies have the infrastructure in place to support computer-based online training. These reports show that there is a large interest in tutoring systems, but also indicate that further research is required to improve the current systems. "While knowledge management is intended to support learning, the discipline itself is still in the process of learning 'how' to manage and leverage knowledge resources (Brown and Massey 1997)."

## **Prior Research**

Intelligent tutoring systems have been examined by a number of researchers. Murray (1996) has been working on authoring tools and criticizes Merrill's work for its lack of authoring tools and its unfriendly user interface. He further advocates against the use of AI technology, because they increase the complexity and thus reduce the usefulness of the system. There is an overall agreement in the literature that expert system/AI support can improve knowledge delivery, however, there is little consensus on how to build such systems. Rosenberg (1987) found that work on intelligent tutoring systems is still in the research stage, and 10 years later, Karagiannidis et al. (1997) still reported that "currently there is a lack of consensus about the characteristics, behavior and essential components of intelligent user interfaces." An intelligent user interface should adapt itself to the learner based on the user profile, the task, and the nature of the application. They present a model for the decision making process that shows how a user interface can adapt based on sets of criteria. However, no implementation or experiments verifying their assumptions are presented.

Another aspect of adaptive interfaces is learner control of content and sequence. Research regarding the extent of learner versus system control is inconclusive. Goforth (1994) found that learner control is more effective than system control. Young (1996) supported this finding only for learners with high self-regulated learning strategies (SRLS), not for others. He reported that system control minimizes the performance difference between low and high levels of SRLS.

The extreme case of full learner control are hypermedia systems. Here, Schroeder and Grabowski (1995) found that novice learners used a fairly passive strategy for moving through the system, not utilizing their selection control and instead following a linear viewing pattern. This problem of *the learner can be lost in countless rounds* is also reported by Frasson and Aimeur (1996). Yang and Moore (1995) criticized that existing hypermedia systems are mostly technology-based without an underlying theoretical basis. However, their implementation of a prototype in HyperCard is fairly limited, because it limits itself to static pages without database or knowledge base support, thus their conclusions can not be generalized.

The type of learning has an impact on access behavior as well. Rieman (1996) showed that no specified sequence is followed in exploratory learning. However, this form of learning was rarely used without a specific task in mind, and users believed it otherwise to be an inefficient way of learning. Individual characteristics such as cognitive style and field dependence also have an impact on the usage of a hypermedia system (Leader and Klein, 1996). Field-dependent individuals view parts of a field as fused, while field-independent individuals experience parts as distinct from the field as a whole. Both differences have an impact

on performance under different conditions, such as the usage of maps versus menus, etc. (Leader and Klein, 1996). Also, promoting exploration in hypermedia systems has a negative impact on screen readability, and using menus instead of embedded hyperlinks changes the usage of the system (Welsh, Murphy et al., 1991). These findings support the addition of a user model to a knowledge base to improve the delivery of the knowledge by using an intelligent user interface that adapts to the user based on the user's characteristics and the type of knowledge that has to be delivered.

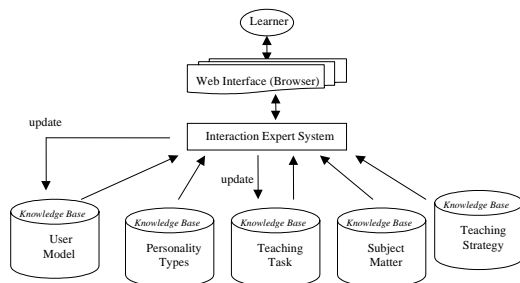
### Prototype of the ITS System

Finding a domain that can be adequately represented by current technology is a fundamental issue (Sangster and Wilson, 1991). They propose that the same principles of decision support applications can be used by computer assisted learning systems. The field of computer programming languages is a very well defined domain and has been used in the past to experiment with knowledge based training systems (e.g. LISP Tutor of Anderson, 1985; ITEM/IP with Turingol, Brusilovski et al. 1982). Therefore, our prototype will use as the subject domain *programming languages*. This also allows the authors to utilize their existing expertise, since both authors are experienced instructors of programming languages. Peer review is used to reduce the impact of designer bias.

Merrill's work on second generation instructional design (Merrill, 1991a, 1991b, 1992a, 1992b, 1993, 1996) lays out the design of a knowledge base for training material and the design of instructional transaction shells. These shells are sets of instructional transactions that are independent of the knowledge bases, but can be applied to them. This is made possible by using an Elaborated Frame Network (EFN) for a formal knowledge representation syntax (Merrill 1993). Processes, entities, and activities are the frame types (classes) of EFN. The following types of transactions are supported in Merrill's model: 1) component transactions: identify, execute, or interpret an instance or class. 2) abstraction transactions: judge, classify, decide, generalize, or transfer an instance or class. 3) association transactions: propagate, analogize, substitute, design, and discover an instance or class. Park and Seidel (1997) criticized frame-oriented computer-based instructional system structures, because the student has "little or no initiative in the instructional process." Merrill's frames, however, support many different kinds of interaction modes: overview, presentation, practices, and assessments, with all these modes under either learner or system sequencing control (Merrill, 1992). An example for an interaction would be to practice the classification of instances of a class, e.g. the *loop* construct, or to interpret a particular sequence of actions, e.g. an *algorithm*. A system based on Merrill's work (1992), enhanced by a user model, a feedback mechanism, personality type knowledge, and teaching task knowledge, is presented in Figure 1.

To make such a system available to the largest number of people possible, using current technology, such a system should be Internet or Intranet based. Browsers support multimedia and hypertext, which can be linked to CGI programs and thus to databases, knowledge bases, and expert systems. We are experimenting at this point with various databases and CLIPS/JESS to see which combination works best for our needs.

Hypermedia applications, which a Web-based system would be, are mostly learner controlled (Yang and Moore, 1995). However, the degree of learner control should depend on the learners familiarity with the topic, the motivation, aptitude, and attitude (Merrill, 1992). Thus, a model of the user should be a part of the knowledge base. An expert system can use the user model and meta-information about the training material, like the best way to present it under certain conditions, time, etc., and information about available user browser/ software and hardware etc. to choose a best instructional delivery method. For example, a user can have a preference for visual representation of training material, but have a 14" monitor with only 16 colors and a slow Internet connection. The expert system would have to weight the factors and decide on how to present a given piece of information. Also, the cognitive style (Jones, 1994) of the learner has to be considered in the interface design and the selection of instructional material.



User Model: Personal knowledge, past performance, cognitive style, personal preferences, etc.  
 Personality Type: Based on Myers-Briggs Type Indicator  
 The Teaching Task: Present in the best way knowledge X to a personality type Y using the teaching strategy Z.  
 Subject Matter: Domain knowledge about programming languages formatted in Merrill's EFN format.  
 Teaching Strategy: Merrill's Instructional Transaction Shells, e.g. overview, present, practice, etc.

**Figure 1. Proposed Layout of the ITS System**  
 (Adapted from (Merrill, 1996).

The ITS prototype merges several technologies: First, a hypermedia knowledge base on the subject domain *programming languages*. Second, an expert system component that a) models users and b) uses information about the users cognitive style, current teaching strategy, and current knowledge material to be learned to adjust the way and content of the information to be displayed to the learner. Third, web based user interfaces for authoring of the different knowledge types (teacher and expert side) and information presentation (learner side). The goal is to create a system that can be utilized on multiple knowledge domains in various ways, from simple information retrieval to personalized learning-on-demand. Since even experts disagree on the best training strategies, we built in a feedback mechanism in our system that learns from past user behavior and continuously improves the way knowledge is presented to the user.

### Future Research

Experiments with students and corporate learners should be conducted to verify the assumption that the system improves learning compared to conventional hypermedia systems. As a refinement, the teaching task could be selected using a neural net as the expert system. We plan on conducting these experiments and will present the results in our next paper.

### References

References are available from the authors.