

# PIModel: A Pragmatic ITS Model Based on Instructional Automata Theory

Jinxin Si, Xiaoli Yue, Cungen Cao, Yuefei Sui

Knowledge Acquisition and Sharing Group  
Key Laboratory of Intelligent Information Processing  
Institute of Computing Technology  
Chinese Academy of Sciences  
Beijing 100080, China  
{jxsi, xlyue, cgcao, yfsui}@ict.ac.cn

## Abstract

It is a vital and challenging issue in AI community to get the “Right Information” to the “Right People” in the “Right Language” in the “Right Timeframe” in the “Right Level of Granularity”. More and more researchers recognized that a well-designed instructional plan in intelligent tutoring systems will open up new possibilities of genuinely intelligent knowledge delivery for future education society. This paper proposes a novel three-layer student-state model and discusses the core elements in the architecture of a practical ITS model based on the theory of instructional automata. The main advantages of instruction automata theory lie in that it can not only generate, regulate, update and implement instructional plans for individualized learner in the efficient and effective way, but also provide a uniform and extensible domain-independent environment for ITS designers and engineers.

## Introduction

With deeper fusion of advanced technology in artificial intelligence and research theory in pedagogical science and cognitive science, intelligent tutoring systems are endowed with more and more attention and energy by researchers and developers, which are distinctive from and more individualized than traditional “one-size-fits-all” approach (Brusilovsky 2001; Brusilovsky 2003). During more than thirty years from 1970 when the system SCHOLAR was developed as an beneficial attempt to build an ITS, a large number of ITSs have successfully been designed, implemented, tested, and approved in varied domains from expertise skills to public education. In recent years, a common agreement about the core elementary models of an ITS seems to be reached in the ITS community, which consists of at least four models, i.e. an expert knowledge model (viz. Domain Model), a pedagogical knowledge model (viz. Pedagogical Model), a model that represents the learner state in the system (viz. User Model) and a user interface model (viz. Human-Computer Interface) (Martens 2003;

Matsuda and VanLehn 2000; Murray and VanLehn 2000; Zapata-Rivera and Greer 2001). Recently, Reddy challenged AI researchers with three open problems (Reddy 2003). One of the problems is “Encyclopedia on Demand”, which Jaime Carbonell calls the “Bill of Rights for the Information Age” (namely, how to get the “Right Information” to the “Right People” in the “Right Language” in the “Right Timeframe” in the “Right Level of Granularity”). From our perspective, this problem could be included in the research area of knowledge design, and there are some common theoretical grounds between knowledge design and intelligent tutoring systems both of which can tailor knowledge “product” to consumer’s requirements (Yue and Cao 2003).

Recently, psychological researches have appreciated that significant improvements in learning ability occur when instructional strategies in content and navigation planning are geared toward the cognitive processing fashion of learners (Mohan 1992; Murray and VanLehn 2000; Wasson. 1992). It is an important issue how to deal with learner control from “tutor authority” to “tutee democracy”, which results in the debate about the locus of control (LOC) between the Computer System / Designer (SYS) and the Learner(L) (Kay 2001; Vassileva and Wasson 1996).

Accordingly, PIModel is presented as a pragmatic ITS model based on the instructional automata theory. The remainder of this paper is organized as follows. In Section 2, we first introduce the method about student state models which is fundamental to provide individualized service to learners. In Section 3, we propose the novel idea about instructional automata theory and concentrate on its architecture. In Section 4, we conclude the paper and raise a few problems for our future research.

## Student State Modeling

User model can help ITSs to select adaptable content and navigation to a given user in efficient and effective way. John Self proposed that during user modeling process there are an “ideal” student and a “real” student, such that the former holds no misconceptions, reasons and learns

rationally, and the latter is naturally less considerate and more uncertain (Self 1994).

Obviously, student states can be depicted by the tuple  $\langle S_s, S_t \rangle$  anytime, where symbol  $S_s$  indicates practical student states from student perspectives and symbol  $S_t$  indicates planned student states from tutor perspectives. In other words, symbol  $S_s$  describes factual learning states which can be attained through pedagogical evaluation during the learner's cognitive process, and the symbol  $S_t$  reflects expected learning states which can be analyzed and inferred through assessing teaching actions and learner response. Both unit exercises and class quizzes need to be considered during the pedagogical evaluation. For example, teaching actions can be "Teach", "Review" and "Remedy", and some variables (e.g. spent time, repetition times, attention focus and facial emotion i.e.) are introduced into learner responses. In Figure 1, we give an explicit and integrated description of the student states from a three-level structure that comprises knowledge conceptual layer, knowledge methodological layer and meta-cognitive layer. At the same time, there are corresponding predictions used to elaborate the functions of these three layers.

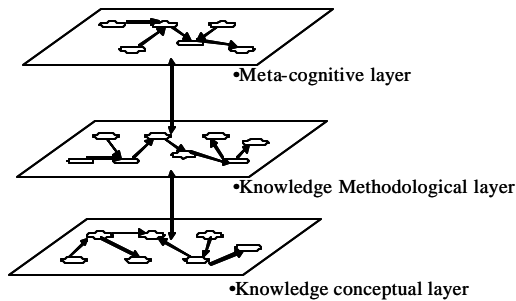


Figure 1. Three-layer Architecture of Student State Model

### Knowledge Conceptual Layer

Given a knowledge concept  $k$ , there are three predications derived from  $k$  in the knowledge conceptual level as follows.

1.  $known(k)$ . It means that a student knows and masters concept  $k$  after the occurrence of instruction about  $k$ .
2.  $unknown(k)$ . It means that a student is still unknowable about concept  $k$  after the occurrence of instruction about  $k$ .
3.  $error(k)$ . It means that a student holds cognitive problem about concept  $k$  after the occurrence of instruction about  $k$ . Furthermore, the prediction  $error(k)$  is classified into two kinds:  $misconception(k)$  and  $missing(k)$ .

- The former denotes that there is a misconception about concept  $k$  which will result in cognitive deviation for a given learner. For example, the concept "Inheritance" is mistaken into the concept "Polymorphism" in language C++, as behaves among many programming novices.

- The latter indicates that for a given student (s)he holds incomplete information about concept  $k$  which maybe results from leaning distraction, and learning and forgetting within relatively long period. For instance, the student cannot remind "the data link layer" among seven layers of the concept "OSI network architecture", which will incur cognitive difficulties to the comprehension of another advanced concept "network communication".

Therefore, for a given student, the whole structure about knowledge concepts (named  $\Omega$ ) is defined as follows:

$\Omega = \{known(k), unknown(k), error(k) \mid k \in K\}$ , where  $K$  is the set which consists of all knowledge concepts for delivery.

### Knowledge Methodological Layer

Given a method  $m$ , there are also three predictions derived from  $m$  in the knowledge methodological layer as follows.

1.  $capable-use(m)$ . For a given method  $m$ , the student is full capable of making use of  $m$  into some applicable scenarios successfully.
2.  $incapable-use(m)$ . For a given method  $m$ , the student is incapable of making use of  $m$  into some applicable scenarios on the premise of knowing the method  $m$ .
3.  $misuse(m)$ . In a given applicable scenario, the student employs some inaccurate method that gives birth to an undesired end-state. It is requisite to point out that the theoretical perspective about human error was developed by James Reason in 1990 (Reason 1990). Reason's definition of error is quoted extensively as below:

"Error will be taken as a generic term that encompasses all those occasions in which a planned sequence of mental or physical activities fails to achieve its intended outcome, and when these failures cannot be attributed to the intervention of some chance agency."

Table 1. Errors' Distinction

Student Error	Is the action appropriate?	Is the action performed correctly?	Is the goal attained?
no error	√	√	√
Mistake	×	√	×
Slip/Lapse	√	×	×

Moreover, Rasmussen proposed the distinction among the three performance-levels corresponding to decreasing levels of familiarity or experience with the environment or task (Rasmussen 1983). These performance levels are labeled skill-based, rule-based and knowledge-based respectively. On the basis of their work, Laughery offered some guidance about the cognitive characteristics, abilities

and limitations of people that have implications in some respects (e.g. error cause and effect, error prevention and correction) (Laughery and Wogalter 1997). Therefore, we can classify the coarsely-grained prediction misuse(m) into three kinds of finely-grained predictions: mistake(m), slip(m) and lapse(m). Table 1 depicts an explicit classification of these predictions from intention perspective. Table 2 illustrates the origins of their causes, prevention mechanism and performance occurrence.

Table 2. Errors' Cause and Effect

Error Type	Cognitive Cause	Prevention Mechanism	Performance Level
Mistake	Mental failures	More training and timely hint	Rule-based or Knowledge-based
Slip	Attentional or perceptual failures	Good display and avoid distractions	Skill-based
Lapse	memorial failures	Memory aids, minimizing time delays	Skill-based

Accordingly, for a given student, the whole structure about knowledge methods (named  $\Psi$ ) is defined as follows:  $\Psi = \{\text{capable-use}(m), \text{incapable-use}(m), \text{misuse}(m) \mid m \in M\}$ , where M is the set which consists of all knowledge methods.

### Meta-Cognitive Layer

Given a cognitive ability c, there are three predictions derived from c in the meta-cognitive layer. These predictions include good(c), average(c) and poor(c). Some psychological experiments argue that the taken granularity of instructional actions (e.g. hint, remedy) for some a learner is heavily dependent on one's meta-cognitive level (Lesgold 1988; Siemer and Angelides 1998).

The taxonomy of cognitive levels was proposed by Bloom and his associates (Bloom et al. 1956): knowledge, comprehension, application, analysis, synthesis and evaluation. Based on their work, Wasson used three types of learning outcomes: fact, analysis, and synthesis (Wasson 1990). We define all cognitive levels by set C as follows:  $C = \{\text{knowledge-ability}, \text{comprehension-ability}, \text{application-ability}, \text{analysis-ability}, \text{synthesis-ability}, \text{evaluation-ability}\}$

Accordingly, for a given student, the whole structure about meta-cognitive levels is defined as follows:

$$\Phi = \{\text{good}(c), \text{average}(c), \text{poor}(c) \mid c \in C\}.$$

### Comparative Operators between Student States

We define some useful operators in order to compare two student-states, which can not only describe the same student from both practical and planned viewpoints, but also depict two different students from the same viewpoint. Among these five comparative operators discussed in the following, the first three operators denote the comparison

between one student-state with another student-state, the last two operators show the subordinate relation between two student-state sets.

1.  $\gg$  The operator indicates that the learning state of the former learner is higher than that of the latter. E.g.  $S_t.\text{good}(c) \gg S_s.\text{poor}(c)$
2.  $\ll$  The operator indicates that the learning state of the former learner is lesser than that of the latter. E.g.  $S_t.\text{poor}(c) \ll S_s.\text{good}(c)$
3.  $=$  The operator indicates that the learning state of the former learner is equal to that of the latter. E.g.  $S_t.\text{capable-use}(m) = S_s.\text{capable-use}(m)$
4.  $\subset$  The operator indicates that the set of learning state of the former learner is subset of that of the latter. It can be formalized by the formula as follows:  
 $\forall c(c \in C) (S_t.p(c) \supseteq S_s.p'(c)) \wedge \exists x(x \in C) (S_t.p(x) \gg S_s.p'(x)) \rightarrow S_s.P'(C) \subset S_t.P(C)$
5.  $\supset$  The operator indicates that the set of learning states of the latter learner is a subset of that of the former. It can be formalized by the following formula:  
 $\forall c(c \in C) (S_t.p(c) \subseteq S_s.p'(c)) \wedge \exists x(x \in C) (S_t.p(x) \ll S_s.p'(x)) \rightarrow S_t.P(C) \supset S_s.P'(C)$

### What's Instructional Automata Theory?

We propose that intelligent tutoring systems can be formalized, analyzed, implemented and evaluated by the uniform intelligent tutoring automaton based on the framework of pushdown automata.

**Definition 1.** Define an intelligent tutoring automaton to be a 6-tuple  $ITA = (Q, \Sigma, \Gamma, q_0, F, \delta)$ , where

1. Q is a knowledge-state set about learners,
2.  $\Sigma$  is a nonempty set about pedagogical tasks,
3.  $\Gamma$  is a task regulation stack set,
4.  $q_0 \in Q$  is the initial learner-knowledge-state,
5.  $F \subseteq Q$  is the set of final learner-knowledge-states, and
6.  $\delta$  is a learner-knowledge-state transition function,  
 $\delta \subseteq (Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma^*) \times (Q \times \Gamma^*)$

The initial learner-knowledge-state can be obtained through learner's self-assessment, pretest and questionnaire. The remainder of this section will concentrate on the descriptions of a knowledge-state set, pedagogical task set, the task regulation set and the final learner-knowledge-state set, which are core elements in the architecture of an intelligent tutoring automaton.

### The Learner-Knowledge-State Set Q

**Definition 2.** Define an admissible learner-knowledge-state is a subset of set Q:

$$Q \subseteq 2^{\Omega \cup \Phi \cup \Psi} \times 2^{\Omega \cup \Phi \cup \Psi} \text{ such that}$$

1. For every state  $q = (q1, q2) \in Q$ , q1 represents the expected state of the learner and q2 denotes the practical state of the learner.

2. Q is legitimate if and only if it satisfies the three conditions below:
  - For every knowledge concept k, there is at most one predication(known(k), unknown(k) or error(k)) which belongs to Q.
  - For every knowledge method m, there is at most one predication(capable-use(m), incapable-use or misuse(m)) which belongs to Q. Some axioms correlate the knowledge method layer with knowledge concept layer, e.g.  $\text{capable-use}(m) \vee \text{incapable-use}(m) \rightarrow \text{known}(k)$ .
  - For every meta-cognitive level c, there is at most one predication(good(c), average(c) or poor(c)) which belongs to Q. At the same time, some axioms associate the meta-cognitive layer with knowledge method layer, e.g.  $\text{good}(\text{knowledge-ability}) \wedge \text{poor}(\text{comprehension-ability}) \rightarrow \text{misuse}(m)$ .

### The Pedagogical Task Set $\hat{\alpha}$

Once the student requirement and the initial student state have been obtained and the expertise in a subject domain is given, the instructional automaton will select an initial efficient teaching strategy based on the student's previous performance. The symbol  $\Sigma$  is used to denote a finite nonempty set and can correspond to the element instruction in educational triple  $\langle \text{curriculum, instruction, assessment} \rangle$  (Dowling and Hockemeyer 1999; Pellegrino, Chudowsky, and Glaser 2001; Pellegrino 2002).

**Definition 3.** The set  $\Sigma$  can be defined as follows:

1.  $\Sigma = \{\text{teach}(\alpha, k), \text{test}(\beta, t), \text{review}(\gamma, k), \text{correct}(\eta, k)\}$ , in which  $\alpha, \beta, \gamma$  and  $\eta$  belong to teaching strategies, k denotes a set of knowledge concepts involved in a pedagogical plan, t represents knowledge item for the assessment of student knowledge states.
2. Teaching strategies (ab. TS) can be classified into TSs with prompt and TSs without prompt. They can be sorted into TSs with guidance, TSs with half-guidance and TSs without guidance.
3. The knowledge item can further be divided into the conceptual item and the inferential item.
4. The instruction action "test" can be classified into class-test (ab. ctest), homework-test (ab. htest), unit-test (ab. utest) and final-test (ab. ftest). Usually, the student's passing-ratio is gained through some objective tests from different aspects.

Two commonly used strategies in curriculum design is the Socratic method and the diagnostic or debugging method (Klein, 1985).

1. The Socratic Method: This method provides the student with questions to guide him to find out his own mistake and thereby modify his conceptions. The process can be formalized as the proposition below:

$(q, Y) = \delta(p, \text{Ctest}(t), Z)$  if  $\text{error}(t) \in q \rightarrow Y \subseteq \{\text{Ctest}(\beta, t), \text{review}(\gamma, k), \text{teach}(\eta, k)\}$

2. The Diagnostic or Debugging Method: In this method the tutor debugs the student's misconceptions and explains why the student made that mistake. Figure 2 depicts this detailed process from the state transition perspective. It can be formalized as the proposition below:

$(q, Y) = \delta(p, \text{test}(\beta, t), Z)$  if  $\text{error}(t) \in q \vee \text{misconception}(t) \in q \vee \text{missing}(t) \in q \rightarrow Y \subseteq \{\text{test}(\beta, t), \text{review}(\gamma, k), \text{correct}(\eta, k)\}$ .

### The Task Regulation Stack Set $\Gamma$

The stack set  $\Gamma$  is equivalent to practical learning history determined by student-assigned detached learning strategies (Mohan, Greer, and Jones 1992) and can be corresponded to the element assessment in educational triple  $\langle \text{curriculum, instruction, assessment} \rangle$ .

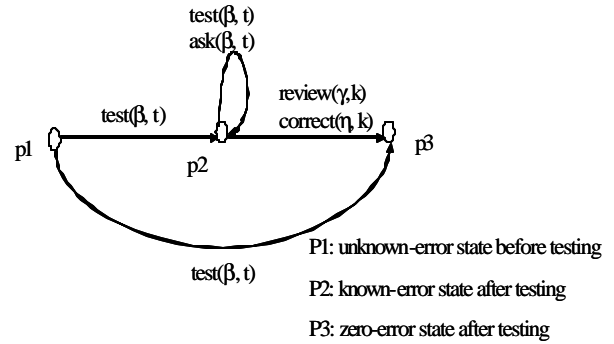


Figure 2. The Diagnostic or Debugging Method

**Definition 4.** Let  $\Gamma$  define as follows:

$\Gamma = \{\langle \text{current goal } g, \text{ pedagogical action } a, \text{ action repetition times } n, \langle \text{action expected time limit } t_d, \text{ practical action spent-time } t_r \rangle, \langle \text{possible errors } e_p, \text{ practical errors } e_r, \text{ error evaluative rules } r_e \rangle \rangle\}$ .

1. The current pedagogical goal  $g$  relates closely with the current instructional task  $t$ :
  - Instructional automata will move to next pedagogical task  $t'$  if  $\text{equal}(g, t) \wedge S_t.\text{error}(k) = \emptyset$
  - Instructional automaton will change the current task regulation stack to new repairing stack (viz. empty-movement) if  $\text{different}(g, t) \vee S_t.\text{error}(k) \neq \emptyset$
2.  $a \in A, A = \{\text{teach}(\alpha, k), \text{test}(t), \text{review}(\beta, k), \text{correct}(k), \dots\}$  (see definition 3)
3. The influence of action repetition times on learner meta-cognitive level is formalized by a logarithmic function.
  - If the instruction action  $a$  is given as  $\text{teach}(\alpha, k)$ , the possibility about  $\text{poor}(\text{comprehension-level})$  will increase more quickly when the repetition times  $n(a)$  is higher.
  - If the instruction action  $a$  is given as  $\text{Ctest}(\alpha, k)$ , the possibility about  $\text{poor}(\text{knowledge-level})$  will increase more quickly when the repetition times  $n(a)$  is higher.

4. The influence of the spent-time tuple  $\langle t_d, t_r \rangle$  on learner meta-cognition level is formalized by a normal distribution function.
  - If the instruction action is given as  $\text{teach}(\alpha, k)$ , the possibility about  $\text{poor}(\text{comprehension-level})$  will increase more quickly when the absolute value between desired spent-time and practical spent-time (viz.  $|t_r(a) - t_d(a)|$ ) is higher.
  - If the instruction action is given as  $\text{test}(\alpha, k)$ , the possibility about  $\text{poor}(\text{application-level})$  will increase more quickly when the absolute value between desired spent-time and practical spent-time (viz.  $|t_r(a) - t_d(a)|$ ) is higher.
5. There are two cases about error triple  $\langle e_p, e_r, e_e \rangle$ , which may result in different choice about next stack.
  - Instructional automata will push down a new stack if  $e_p \subset e_r$ .
  - Instructional automata will push down a previous stack again if  $e_p \not\subset e_r$ .

### The Final Learner-Knowledge-State Set F

There are some classifications in the final learner-knowledge-state set.

1. The final state  $f$  is a final tutor-satisfied state if none of  $\text{error}(k)$ ,  $\text{unknown}(k)$ ,  $\text{incapable-use}(m)$ ,  $\text{misuse}(m)$  and  $\text{poor}(c)$  exists in  $f \wedge S_s(p) \supseteq S_t(p)$ .
2. The final state  $f$  is a final tutee-satisfied state if  $f$  is a final tutor-satisfied state  $\vee (S_t(p) - S_s(p) \neq \emptyset \wedge S_s(p). \text{passing-ratio} \geq \theta \wedge S_s(p). \text{spent-time} \leq \tau \wedge S_t(p). \text{repetition-times} \leq \rho)$ .
3. The final state  $f$  is a final tutor-unsatisfied state if at least one of  $\text{error}(k)$ ,  $\text{unknown}(k)$ ,  $\text{incapable-use}(m)$ ,  $\text{misuse}(m)$  and  $\text{poor}(c)$  exists in  $f \vee S_s(p) \not\supseteq S_t(p)$ .
4. The final state  $f$  is a final tutee-unsatisfied state if  $f$  is a final tutor-unsatisfied state  $\wedge (S_t(p) - S_s(p) \neq \emptyset \wedge S_s(p). \text{passing-ratio} < \theta \vee S_s(p). \text{spent-time} > \tau \vee S_t(p). \text{repetition-times} > \rho)$ .
5. The final state  $f$  is a final tutor-acceptable state if the values for all variables in  $S_s(p)$  and  $S_t(p)$  are deterministic and consistent with pedagogical constraints.
6. The final state  $f$  is a final tutee-acceptable state if the values for all core variables in  $S_s(p)$  and  $S_t(p)$  are deterministic and consistent with pedagogical constraints.
7. The final state  $f$  is a final dangerous state if there is at least one variable in  $S_s(p)$  or  $S_t(p)$  whose value is uncertain.
8. The final state  $f$  is a final warning state if there is at least one variable in  $S_s(p)$  or  $S_t(p)$  whose value is inconsistent with the pedagogical constraints although the values for all their variables are deterministic. For instance, although the student passed the examination, his spent time always exceeds expected time limit greatly.

### Conclusion

The student state model and the core elements in the architecture of a practical ITS model based on instructional automata theory were highlighted in this paper. The main advantages attributed to the instructional automata lie in that it can generate, regulate, update and implement instructional plans in an efficient and effective way, which are tailored to users' static features and dynamic requirements. Moreover, the instructional automata theory provides a unified and extendable domain-independent environment to ITS designers and engineers.

In our future work, instruction initiative will be further taken into account in instructional automata, which can be classified into tutee initiative and tutor initiative. On the basis of instruction initiative, we proposed that the instructional automata can also be classified into two kinds: tutor-centered IA and tutee-centered IA, which strongly affect the granularity of educational delivery during learning. In addition, some measures should be taken in order to deal with the failures during tutoring and evaluation process, as result from inappropriate content and navigation, uncertainty regarding the estimate of the student's knowledge. Therefore, an extension of instructional automata theory that accounts for these problems needs to be designed and developed.

### Acknowledgements

This work is supported by the Natural Science Foundation (#60073017 and #60273019), and the Ministry of Science and Technology (#2001CCA03000 and #2002DEA30036).

### References

- Bloom, B. S.; Englehart, M. D.; Furst, E. J.; Hill, W. H.; and Krathwohl, D. R. eds. 1956. *A Taxonomy of Educational Objectives: Handbook I. Cognitive Domain*. New York: David McKay.
- Brusilovsky, P. 2001. Adaptive hypermedia. *User Modeling and User Adapted Interaction, Ten Year Anniversary Issue* (Alfred Kobsa, ed.) 11 (1/2), 87-110.
- Brusilovsky, P. 2003. Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology*, 34 (4), 487-497.
- Dowling, C.; and Hockemeyer, C. 1999. Automata for the assessment of knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 13, 3, 451-461.

- Kay, J. 2001. Learner control, *User Modeling and User-Adapted Interaction, Tenth Anniversary Special Issue*, 11(1-2), Kluwer, 111-127.
- Klein, M.F. 1985. Curriculum Design. *International Encyclopedia of Education: Curriculum Studies. Volume II*. Husen, T. and Postlethwaite editors, Pergamon Press: Oxford England, pp. 1163-1170.
- Laughery, K.R.; and Wogalter, M. S. 1997. Warnings and risk perception. In Salvendy (Ed.), *Handbook of Human Factors and Ergonomics, Second Edition*.
- Lesgold, A. 1988. Towards a theory of curriculum for use in designing intelligent instructional systems. In Mandl, H. and Lesgold, A. (eds), *Learning issues for Intelligent Systems*. Berlin, Springer Verlag.
- Martens, A. 2003. Centralize the Tutoring Process in Intelligent Tutoring Systems. In Proc. of *the 5th International Conference on New Educational Environments, ICNEE'03*, Lucerne, Switzerland, 26.-28. May.
- Matsuda, N; and VanLehn, K. 2000. Decision Theoretic Instructional Planner for Intelligent Tutoring Systems, in Proc. for *Workshop on Modeling Human Teaching Tactics and Strategies*, ITS2000, pp.72-83, 2000.
- Murray, R. C.; and VanLehn, K. 2000. DT Tutor: A decision-theoretic, dynamic approach for optimal selection of tutorial actions. In *Proceedings of ITS 2000*. Berlin: Springer-Verlag.
- Mohan, P.; Greer, J. E.; and Jones, M. L. 1992. Enhancing the Instructional Capabilities of Intelligent Tutoring Systems; In *Proceedings of Intelligent Tutoring Systems (Second International Conference)*, Springer Verlag, June, 532-539.
- Pellegrino, J. 2002. Understanding How Students Learn and Inferring What They Know: Implications for the Design of Curriculum, Instruction, and Assessment, Paper presented at NSF Instructional Materials Development Conference, February.
- Pellegrino, J. W.; Chudowsky, N.; and Glaser, R. 2001. *Knowing what students know: The science and design of educational assessment*. Washington, DC: National Academy Press.
- Reason, J. 1990. *Human Error*. Cambridge, Cambridge University Press.
- Rasmussen, J. 1983. Skills, rules, knowledge: Signals, signs, and symbols and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics*, 13, 257-267.
- Reddy, R. 2003. Three open problems in AI. *JACM* 50(1): 83-86.
- Self, J.A. 1994. Formal approaches to student modelling, in J.E. Greer and G.I. McCalla (eds.), *Student Modelling: the Key to Individualized Knowledge-Based Instruction*, Berlin: Springer-Verlag.
- Siemer, J ; and Angelides, M. C. 1998. Towards an Intelligent Tutoring System Architecture that Supports Remedial Tutoring. *Artificial Intelligence Review* 12(6): 469-511.
- Vassileva J.; and B. Wasson. 1996. Instructional Planning Approaches: from Tutoring towards Free Learning. *Proceedings of Euro-AIED'96*, Lisbon, Portugal, 30.9.-2.10. 1-8.
- Wasson, B. 1990. Determining the focus of Instruction: content planning for intelligent tutoring systems. Ph.D. Thesis, Department of Computer Science, University of Saskatchewan, Saskatoon, Canada.
- Wasson, B. 1992. PEPE: A computational framework for a content planner. In S.A. Dijkstra, H.P.M Krammer & J.J.G. van Merriënboer (Eds.) *Instructional Models in Computer-based learning environments*. NATO ASI Series F, Vol 104, 153-170. New York: Springer-Verlag.
- Yue X.; and Cao C. 2003. Knowledge Design. In *Proceedings of International Workshop on Research Directions and Challenge Problems in Advanced Information Systems Engineering*, Japan, Sept.
- Zapata-Rivera, J.; and Greer J. 2001. SMODEL server: student modelling in distributed multi-agent tutoring systems. *Proceedings of the 10th International Conference on Artificial Intelligence in Education*, pages 446-455, San Antonio, Texas, May 19-23.