

# INTERACTIVE EVENTS

Amsterdam, 18-22 July, 2005

Editor: **Lora Aroyo**



12th International Conference on Artificial  
Intelligence in Education, Amsterdam,  
the Netherlands



12th International Conference on Artificial Intelligence in Education  
AIED'05

## Interactive Events

July, 2005  
Amsterdam, The Netherlands

## Table of Contents

Interactive-Event: Personalized e-Learning Services for the Semantic Web <i>Nicola Henze</i>	1
Rewards Mechanism for Sustainable Online Learning Communities <i>Julita Vassileva</i>	5
Acharya: Tutors for C++ Programming <i>Amruth Kumar</i>	7
The Tactical Language Training System <i>W. Lewis Johnson and Hannes Högni Vilhjalmsson</i>	11
Rapid ITS Prototyping with Ontologies in WETAS <i>Brent Martin, Antonija Mitrovic and Pramudi Suraweera</i>	15
OntoAIMS: Semantically-aware and User-adaptive Learning Content Management <i>Lora Aroyo and Vania Dimitrova</i>	19
Exploring meta-cognitive tutoring by the Help Tutor: An Interactive Event <i>Vincent Aleven, Bruce McLaren, Ido Roll and Ken Koedinger</i>	21
Human Centred Design for Interactive Educational Software <i>Rosemary Luckin and Geraldine Fitzpatrick</i>	25
Interactive Demonstration of a Generic Tool to Browse Tutor-Student Interactions <i>Jack Mostow, Joseph Beck, Hao Cen, Evandro Gouvea and Cecily Heiner</i>	29
A Tutoring System to Practice Theorem Proving <i>Sicco Verwer, Mathijs de Weerd and Jonne Zutt</i>	33
Elements of ITS development technology and evaluation of its effectiveness <i>Ildar Galeev, Oleg Kolosov and Alexander Filyaev</i>	37



# Interactive-Event: Personalized e-Learning Services for the Semantic Web

Nicola Henze <sup>a,1</sup>

<sup>a</sup> *ISI – Semantic Web Group,  
University of Hannover & Research Center L3S*

**Abstract.** During this Interactive Event, we demonstrate how to create personalized e-Learning courses based on Semantic Web technologies. Starting with a set of learning objects, we create a machine-readable, semantic description of an e-Learning course. We select appropriate personalization rules for this course, and see how these rules can be maintained by Web-services. We run the generated course in the Personal Reader Framework for personalized e-Learning in the Semantic Web.

**Keywords.** personalization for the semantic web, personalization services, e-learning architectures

## 1. Introduction

Personalization for the (Semantic) Web is still in its infancy. We are lacking flexible and re-usable personalization strategies which can be applied in various but similar contexts. One approach to overcome this problem is working towards personalization plug-ins: Services, which offer a certain personalization strategy, e.g. creating a guided tour, or recommending information, or annotating materials, etc. [5].

Within the Personal Reader project, we have developed a framework for designing, implementing, and maintaining *personalized* Web Content Readers<sup>2</sup>. The Personal Reader framework makes use of recent Semantic Web technologies for realizing a service-based environment for implementing and accessing personalization services. Several, distributed services - for providing the user interface, for mediating between user requests and available personalization services, for user modeling, for providing personal recommendations and context information, et cetera, form the core of the Personal Reader framework. Prototypes of Personal Readers for e-Learning have been realized for the topics “programming in Java” and “Semantic Web” [3]. An authoring component for including new courses and creating new Personal Reader instances is available and will be demonstrated at the interactive event. As a proof-of-concept of the underlying architecture of the Personal Reader framework, a Personal Reader for browsing scientific publications has been realized [1].

---

<sup>1</sup>Correspondence to: Nicola Henze, ISI - Semantic Web Group, University of Hannover & Research Center L3S, Appelstr.4, D-30167 Hannover Tel.: +49 511 762 19716; Fax: +49 511 762 19712; E-mail: henze@l3s.de

<sup>2</sup>This work has partially been supported by the European Network of Excellence REWERSE - Reasoning on the Web with Rules and Semantics ([www.rewerse.net](http://www.rewerse.net)).

## 2. Creating Personal Readers for e-Learning

The Personal Reader Framework offers ready-to-use *Personalization Services* for e-Learning. These Personalization Services realize some of the adaptation techniques from the area of adaptive educational hypermedia (For an overview on methods and techniques of adaptive hypermedia, we refer the reader to [2]. A logical formalism for adaptive educational hypermedia is given in [6]). Authoring is a very critical issue for successfully realizing adaptive educational hypermedia systems. As a guideline for our work, we established the following rule:

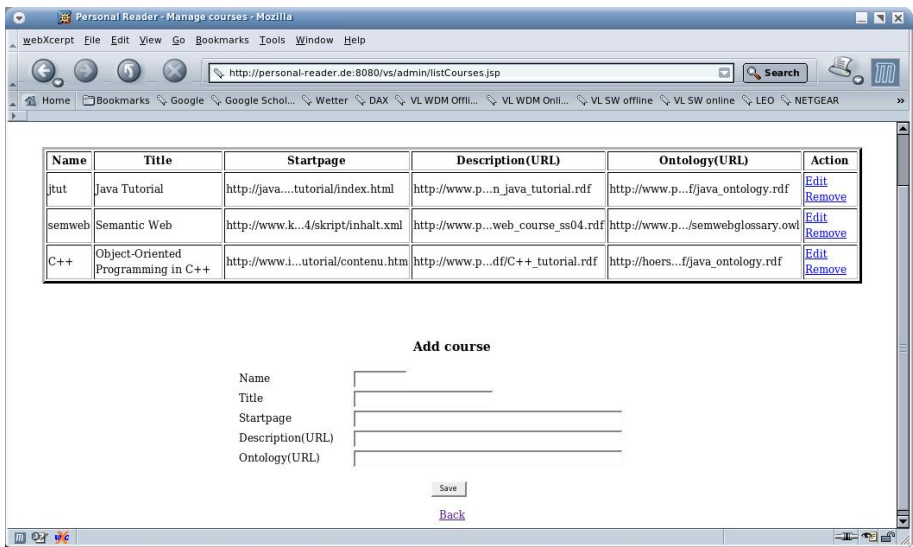
Learning objects, course description, domain ontologies, and user profiles *must* be annotated according to existing standards (for details please refer to [4]). The flexibility must come from the personalization services which must be able to reason about these standard-annotated learning objects, course descriptions, etc.

For example, to create a new Personal Reader for an e-Learning course, the author of the course has to provide a metadata description of the new course in the language of RDF (Resource Description Framework [8]). Learning objects in the course are annotated according to recent standards for e-Learning objects like LOM ([7]). The following code gives an example of such a course description for a course on Java Programming<sup>3</sup>.

```
<rdf:RDF xml:lang="en"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms#"
  xmlns:lom="http://ltsc.ieee.org/2002/09/lom-base#"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description rdf:about="http://java.sun.com/.../tutorial/index.html">
    <rdf:type rdf:resource="http://ltsc.ieee.org/2002/09/lom-educational#Book"/>
    <dc:title>The Java Tutorial (SUN)</dc:title>
    <dc:creator>
      <lom:entity> <vCard:FN>M. Campione</vCard:FN> </lom:entity>
      <lom:entity> <vCard:FN>K. Wallrath</vCard:FN> </lom:entity>
    </dc:creator>
    <dcterms:hasPart>
      <rdf:Seq> <rdf:li rdf:resource=".../tutorial/java/index.html"/> </rdf:Seq>
    </dcterms:hasPart>
  </rdf:Description>
  ...
  <rdf:Description rdf:about="http://.../tutorial/java/concepts/message.html">
    <rdf:type rdf:resource="http://ltsc.ieee.org/2002/09/lom-educational#LO"/>
    <dc:title>What Is a Message?</dc:title>
    <dc:subject rdf:resource=
      "http://www.personal-reader.de/rdf/java_ontology.rdf#OO_Methods"/>
    <dcterms:isPartOf rdf:resource="http://.../tutorial/java/concepts/index.html"/>
  </rdf:Description>
  ...
</rdf:RDF>
```

<sup>3</sup>The complete description is available at [http://www.personal-reader.de/rdf/sun\\_java\\_tutorial.rdf](http://www.personal-reader.de/rdf/sun_java_tutorial.rdf)



**Figure 1.** Screenshot of the administration interface of the Personal Reader, showing how to integrate new course materials

The administration component (see Figure 1) of the Personal Reader framework provides an author interface for easily creating new instances of course-Readers: Course materials which are annotated according to LOM (or some subset of it), and which might in addition refer to some domain ontology, can immediately be used to create a new Personal Reader instance which offers all the personalization functionality which is - at runtime - available in the personalization services.

An example on determining more detailed learning objects on some given learning object is given in the following rule: Let LO denote some given learning object, and LO\_DETAIL denote the detailed learning objects we are looking for. LO\_DETAIL fulfills our requirements, if it covers some learning concepts C\_DETAIL which are details of those learning concepts covered in LO, or if LO\_Detail is a subconcept of LO in the course structure. In the TRIPLE [9] language this rule looks as follows:

```
FORALL LO, LO_DETAIL detail_learningobject(LO, LO_DETAIL) <-
  ( EXISTS C, C_DETAIL(detail_concepts(C, C_DETAIL)
    AND concepts_of_LO(LO, C)
    AND concepts_of_LO(LO_DETAIL, C_DETAIL)))
  OR (upperlevel(LO_DETAIL, LO)) .
```

Rules like the above are maintained within the personalization services. At runtime, each currently in the Personal Reader Framework registered Personalization Service receives a request (in RDF) with information about the user, the page this user is currently visiting, etc., and generates e.g. personal recommendations for this user. These recommendations are coded in RDF, too, and passed back to the Personal Reader framework. Figure 2 shows a screenshot of the Personal Reader for the Java programming course.

The current state of the Personal Reader project can be followed at [www.personal-reader.de](http://www.personal-reader.de), where all the realized prototypes are available, too.

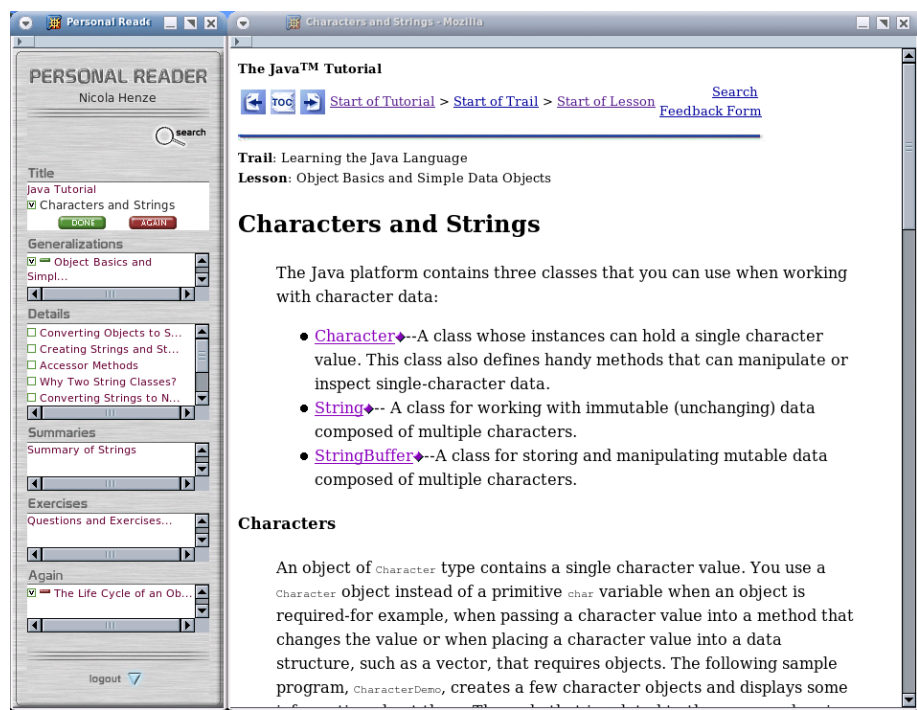


Figure 2. Screenshot of the Personal Reader for Java programming

References

[1] Robert Baumgartner, Nicola Henze, and Marcus Herzog. The Personal Publication Reader: Illustrating Web Data Extraction, Personalization and Reasoning for the Semantic Web. In *European Semantic Web Conference ESWC 2005*, Heraklion, Greece, May 29 - June 1 2005.

[2] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2-3):87–129, 1996.

[3] Nicola Henze. Personal Readers: Personalized Learning Object Readers for the Semantic Web. In *12th International Conference on Artificial Intelligence in Education, AIED'05*, Amsterdam, The Netherlands, July 2005.

[4] Nicola Henze, Peter Dolog, and Wolfgang Nejdl. Reasoning and ontologies for personalized e-learning. *Educational Technology & Society*, 7(4), 2004.

[5] Nicola Henze and Matthias Kriesell. Personalization Functionality for the Semantic Web: Architectural Outline and First Sample Implementation. In *1st International Workshop on Engineering the Adaptive Web (EAW 2004)*, Eindhoven, The Netherlands, 2004.

[6] Nicola Henze and Wolfgang Nejdl. A logical characterization of adaptive educational hypermedia. *New Review of Hypermedia*, 10(1), 2004.

[7] LOM: Draft Standard for Learning Object Metadata, 2002. <http://ltsc.ieee.org/wg12/index.html>.

[8] Resource Description Framework (RDF): Concepts and abstract syntax, February 2004.

[9] Michael Sintek and Stefan Decker. TRIPLE - an RDF Query, Inference, and Transformation Language. In *International Semantic Web Conference (ISWC)*, Sardinia, Italy, 2002.

# Rewards Mechanism for Sustainable Online Learning Communities

Julita Vassileva

*Computer Science Department, University of Saskatchewan,  
Saskatoon, Canada S7N 5C9, [jiv@cs.usask.ca](mailto:jiv@cs.usask.ca)*

## Description of the Interactive Event

The interactive event will engage the participants in the Comtella peer-to-peer community to share links to images they find on the Web and to rate the images found by other participants. Comtella is designed to encourage user participation through a built-in incentive mechanism to create a competitive, game-like experience. After a period of using the system, the results will be summarized and the participants will be asked to share their experience about what motivated them to contribute as much as they did. The participants will be asked to find ways to game the system and to test the robustness of the rewards mechanism. The interactive event will conclude with discussion of the importance of designing a good mechanism (rules of the game) on the sustainability of the community.

## Underlying Technology

We developed an incentive mechanism based on theories from social psychology (social comparison, reciprocation) to encourage users to contribute resources. The users can earn a higher membership level, if they perform actions that help the community: contributing new resources, rating resources, and caring about the quality of resources and ratings that they contribute. Each membership level rewards the user with particular interface appearance, and better social visibility (shown in visualization based on a star-sky metaphor) related to the user's status in the community.

To ensure self-moderation of the quality and quantity of resources and in this way to achieve sustainability of the community, the incentive mechanism was augmented with a user- and community-based adaptation. Similar to Slashdot, users with high membership level gain more power to rate contributions of other users. Users are encouraged to rate contributions by rewarding each act of rating with c-points (a virtual currency) that can be used to increase the visibility of the rater's own contributions.

The incentive mechanism has been implemented in a web-based system for sharing links to class-related resources. It has been evaluated for two consecutive years in a fourth year computer science class on Ethics and Information technology. A detailed description of this mechanism and the evaluation results is presented in a paper by Ran Cheng and Julita Vassileva in this volume.



# Acharya: Tutors for C++ Programming

Amruth KUMAR

*Ramapo College of New Jersey,*

*505 Ramapo Valley Road, Mahwah, NJ 07430, USA*

*1 (201) 684 7712*

*amruth@ramapo.edu*

**Abstract.** We have been developing problem-solving tutors on programming topics. The tutors use model-based representation of the domain. Therefore, they can automatically solve the problems that they present. The tutors use a two-stage algorithm to automatically generate an explanation of the correct solution. The student model uses learning objectives as its ontological primitive. The problems are also indexed by learning objectives. The tutors use a rule-based algorithm to adapt the problem sequence based on learning objectives. The tutors deal with analysis problems – debugging and predicting the output of programs. They use reified interfaces to reinforce the use of mental models for both these types of problems. Our evaluations show that the tutors promote student learning; and that they are more effective with explanation than without.

## 1. Introduction

We have been developing tutors to help students learn programming language concepts by solving problems. To date, we have developed tutors on arithmetic and relational expression evaluation, selection, counter-controlled loops, logic-controlled pre-test loops, pointers in C++, parameter passing mechanisms, scope concepts and their implementation, and classes. The tutors adaptively present problems to the learner, grade the learner's answer against the correct answer that they automatically generate, provide a step-by-step explanation of the correct answer, and track the student's performance. They address analysis in Bloom's taxonomy [1], i.e., problems on debugging programs and predicting the output of programs. They are designed to be used as complements to the traditional programming projects, as recommended by the whole language approach [2].

## 2. Domain Modelling

The tutors use model-based representation of the domain [3]. In model-based representation, a model of the problem is first constructed, e.g., a model of the C++ program. This model is simulated to generate its correct behavior, e.g., the output of the program. The correct behavior is compared with the behavior predicted by the student and the differences may be used to hypothesize structural discrepancies in the mental model of the student.

As is evident, one advantage of using a model-based representation of the domain is that the tutor can generate answers for problems on its own, i.e., the domain model doubles as the runnable expert module. Therefore, the tutor can solve problems entered by the student, not just the problems encoded by the instructor.

We use component-ontological representation [4,5], wherein, the model of a device is represented in terms of its components rather than their states. Such representation is modular, reusable and scalable. The domain model can be built incrementally, e.g., the various tutors on

C++ programming share the same domain model, and we have been building tutors on new topics by adding additional programming constructs to the domain model.

### 3. Problem Generation

The tutors generate problems as randomized instances of parameterized problem templates. Every instance of a template is a new problem and no two problems are identical. This enables our tutors to present different instances of a template to different users at a given time (to prevent plagiarism), or to the same user at different times (for test-re-test).

We have identified a set of learning objectives for each programming topic. These learning objectives serve as the ontological primitives of the student model. The problem templates themselves are indexed by these learning objectives. The tutors adapt the problem sequence [6] by using a rule-based expert system to select the next problem template based on the learning objectives that the student has not yet satisfied.

### 4. Feedback Generation

We have developed a two-stage algorithm that uses reflection to automatically generate the step-by-step explanation of the execution of a program [7]. Therefore, the explanation need not be meticulously hand-crafted for each problem. The tutors can even generate explanation for problems entered by the student.

In the two-stage algorithm, each component in the model is responsible for generating the explanation relevant only to its behavior. This makes the explanation generation algorithm scalable. Despite this distributed nature of generation, the resulting explanation is coherent and effective [8].

Figure 1 shows a snapshot of the explanation provided by the tutor on C++ pointers. The left panel displays the problem, and the right panel displays the feedback, including the step-by-step explanation automatically generated by the tutor.

### 4. User Interface

Our tutors deal with analysis problems – debugging and predicting the output of programs. We have developed reified user interfaces for both these types of problems [9]. For debugging problems, the user must identify 1) the line of code that is suspected to have a bug; 2) the program object to which the bug applies, and 3) the nature of the bug, *in that order*. For problems where the student must predict the output of a program, the user must identify the line of code before identifying the output generated by that line of code. These user interfaces promote the use of mental models for solving problems.

Figure 2 shows a snapshot of the reified user interface of the tutor on C++ pointers. Note that the drop-down menu for line 7 displays the program objects on that line, i.e., `firstPointer` and `depth`, followed by the types of possible errors for the chosen object, i.e., `firstPointer`.

### 4. Evaluation

We have evaluated our tutors over multiple courses/semesters/institutions using a pre-test-practice-post-test protocol. Some of the results of our evaluations include: (i) the tutors help students learn programming concepts; (ii) they are more effective when they explain the step-by-step execution of the program; (iii) women learn from the tutors just as well as men; (iv) the self-



confidence of the class improves with the use of the tutors; and (v) women may be more receptive to using online tutors than men.

#### 4. Future Work

We are currently working on the following features: 1) rule-based adaptation of feedback; 2) the use of concept maps for situated learning and navigation; 3) automated allocation of partial credit for programming problems; 4) generation of synthesis problems; 5) model-based visualization of program execution; 6) Hidden Markov models to determine whether a student has satisfied a learning objective; and 7) Bayesian networks and inferential student modeling.

#### 5. Acknowledgements

Partial support for this work was provided by the National Science Foundation's Educational Innovation Program under grant CNS-0426021.

#### References

- [1] Bloom, B.S. and Krathwohl, D.R.: Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain, New York, Longmans, Green (1956).
- [2] Mann, P., Suiter, P., and McClung, R.: A Guide for Educating Mainstream Students. Allyn and Bacon, 1992.
- [3] Kumar, A.N.: Model-Based Reasoning for Domain Modeling in a Web-Based Intelligent Tutoring System to Help Students Learn to Debug C++ Programs. *Proceedings of Intelligent Tutoring Systems (ITS 2002)*, LNCS 2363, Biarritz, France, June 5-8, 2002, 792-801.
- [4] Kumar, A.N. and Upadhyaya, S.J.: Component Ontological Representation of Function for Reasoning About Devices. *Artificial Intelligence in Engineering*, Vol 12(4), (1998) 399-415.
- [5] Kumar, A.N. and Upadhyaya, S.J.: Function-Based Candidate Discrimination During Model-Based Diagnosis. *Applied Artificial Intelligence*, An International Journal, Vol. 9(1), (1995) 65-80.
- [6] Kumar, A.N., Rule-Based Adaptive Problem Generation in Programming Tutors and its Evaluation, Workshop on Adaptive Systems for Web-Based Education: Tools and Reusability, 12th International Conference on Artificial Intelligence in Education (AIED 2005), Amsterdam, The Netherlands, 7/18-22/2005.
- [7] Kumar, A.N. Generation of Demand Feedback in Intelligent Tutors for Programming. *Advances in Artificial Intelligence*, Ahmed Tawfik and Scott Goodwin (eds.), *Proceedings of The Seventeenth Canadian Conference on Artificial Intelligence (AI 04)*, London, Ontario, Canada, 5/17-19/2004, *Lecture Notes in Artificial Intelligence* 3060, Springer, 444-448.
- [8] Kumar, A.N., Learning Programming by Solving Problems, in *Informatics Curricula and Teaching Methods*, L. Cassel and R.A. Reis ed., Kluwer Academic Publishers, Norwell, MA, 2003, 29-39, 29-39.
- [12] Kumar, A.N., A Reified Interface for a Tutor on Program Debugging, *Proceedings of Third IEEE International Conference on Advanced Learning Technologies (ICALT 2003)*, Athens, Greece, 7/9-11/2003, 190-194.

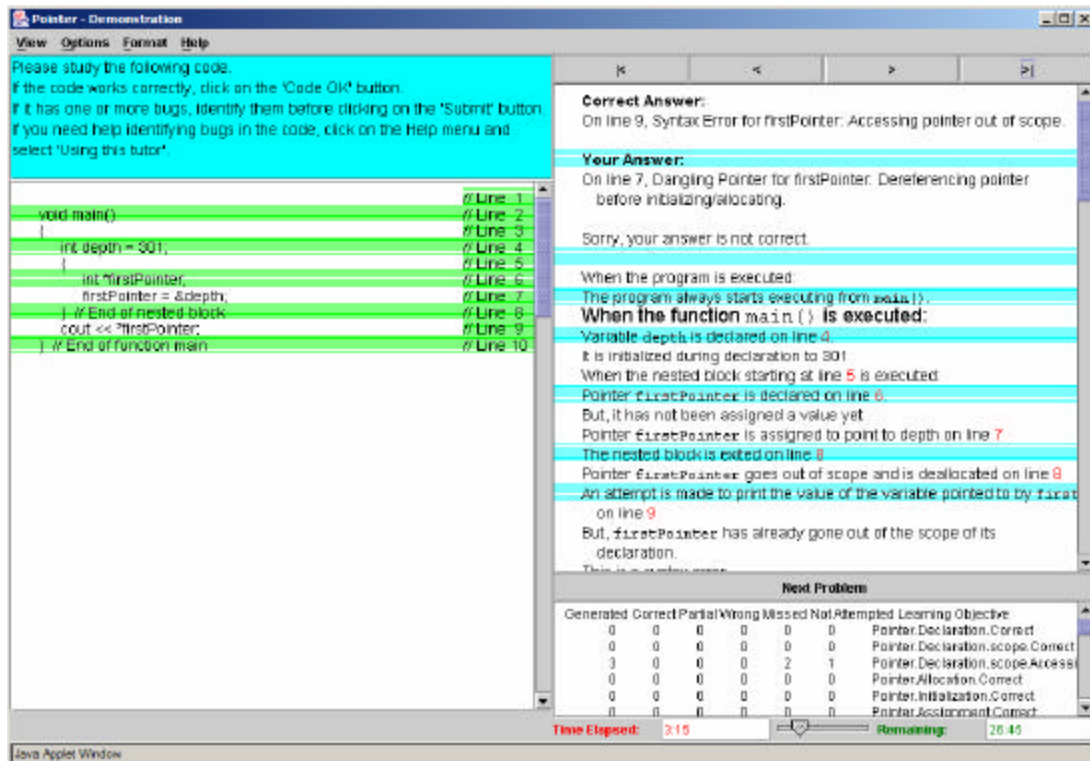


Figure 1: Snapshot of the explanation generated by the tutor on C++ pointers

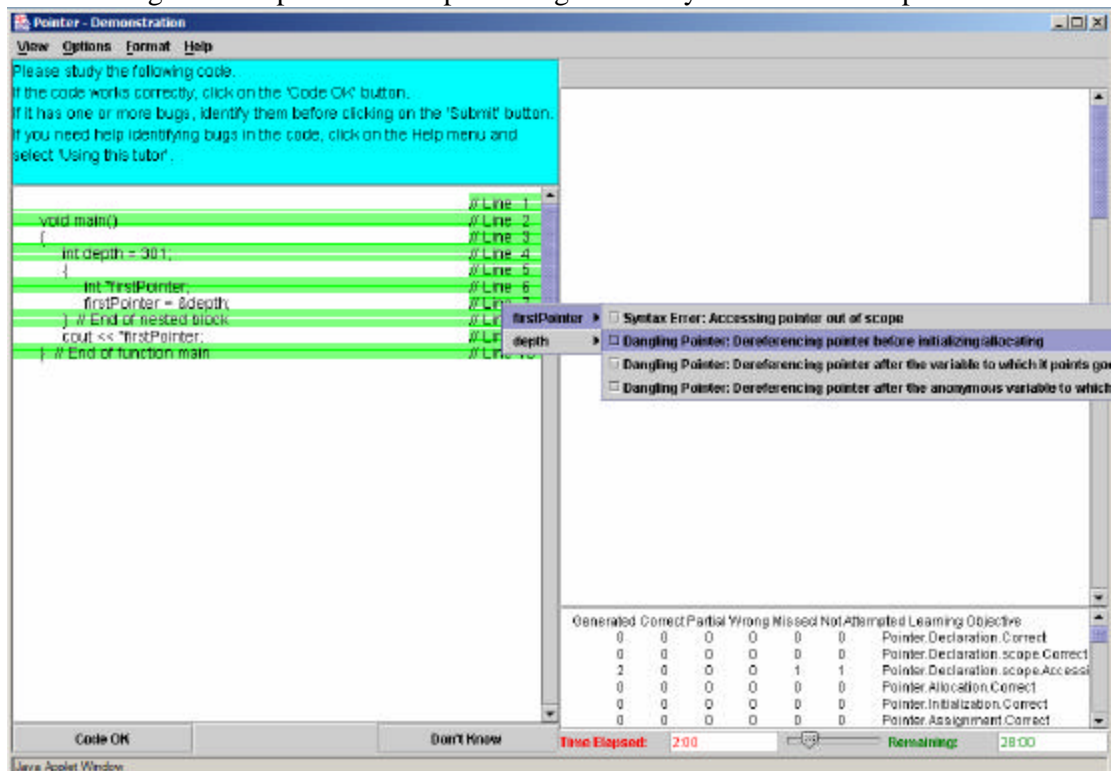


Figure 2: Snapshot of the reified interface of the tutor on C++ pointers

# The Tactical Language Training System

W. Lewis Johnson and Hannes Vilhjálmsón

*Center for Advanced Research in Technology for Education (CARTE)*

*USC/Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292*

**Abstract.** Participants will have the opportunity to interact with the Tactical Language Training System (TLTS), an interactive program that supports rapid acquisition of foreign language and cultural skills. TLTS integrates task-oriented language instruction, intelligent tutoring in spoken communication, and AI-enhanced interactive gaming. Trainees learn the skills necessary to carry out a postwar civil affairs mission, in which they must enter a town or village, establish contact with the local people, meet with the local leader of the district, and arrange for postwar reconstruction. They carry out this mission in a simulated game world, communicating with nonplayer characters using a combination of speech and gesture, and accompanied by a virtual aide who can assist the learner if they run into difficulties, and then give feedback afterwards. The learner can go through a set of interactive exercises that focus on the communication skills needed to carry out the mission, and which the learner gets feedback on their language usage, including pronunciation. The feedback of the virtual tutor is based upon the tactics employed by human language teachers, and implements motivational tutorial tactics. Two training systems have been developed so far using the TLTS: Tactical Levantine Arabic and Tactical Iraqi Arabic. A third, Tactical Pashto, is under development.

## 1. Introduction

The Tactical Language Training System is an intelligent learning environment designed to help people rapidly to acquire basic knowledge of foreign languages and cultures. It incorporates a number of design and implementation features that are innovative and are likely to be of interest to the AIED community.

- The program is designed to promote spoken conversational proficiency. Very few intelligent learning environments support foreign language learning; even fewer aim at spoken proficiency, and of these even fewer are presented at AIED conferences. It is likely that many AIED attendees will be interested in learning how AIED techniques apply to a different domain, a domain that is likely to be of interest to many participants at an international conference.
- The program employs game design techniques. Interactive games can be an effective vehicle for promoting learning, and for motivating students to learn on their own. The potential of games for learning may be underappreciated in the AIED community, as evidenced by the fact that there is no mention of games in the AIED 2005 call for papers. AIED attendees are likely to be interested in learning how game design techniques can promote learning, and understanding how they differ from multimedia environments that do not follow game design principles.
- The program also employs intelligent tutoring techniques. The program has an explicit model of the communicative skills being trained, and the tasks in which the learner is seeking to employ those skills. It also has a model of the learner's degree of proficiency in those skills. We are applying a model tracing approach to learner modelling, based in part Beck and Sison's adaptation of the classical model tracing to speech-enabled tutoring systems.[1]

- The program employs a multimodal interface. Users can speak on behalf of their game character, and choose gestures for their character to perform in the game world. The characters in the game, including the virtual tutor, respond using a combination of speech and gesture. Speech recognition is difficult in ordinary circumstances. The speech recognition task for TLTS is particularly challenging because (1) the speech is spoken by language learners, and is therefore full of errors and disfluencies; (2) the TLTS is designed to detect and diagnose common learner language errors.
- The intelligent tutoring component is designed to address motivational and affective aspects of learner state, not just cognitive state. Many language learners, particularly adults, lack confidence in their abilities, and are afraid of embarrassment. The intelligent tutoring component of the TLTS employs motivational tactics, designed to address motivational problems and counteract negative affectivity. These are modelled on the techniques employed by expert language teachers [3], which have been found to have a similar function to motivational tutoring tactics described by Lepper and his colleagues [4]. Its techniques follow principles of politeness theory [2] and social interaction, as described by Reeves and Nass [5]. In future work we plan to estimate the motivational state of the learner in real time, and adapt feedback accordingly.
- The program is built on top of the Unreal Tournament game engine. Game engines offer a number of advantages to builders of interactive environments, including interactive learning environments. Attendees may be interested in learning how to take advantage of such tools in their own work. Both immersive 3D worlds and multimedia interfaces can be built as “mods” to game worlds and controlled by external agent “minds.”
- Creating learning environments such as these requires the collaboration of people with a range of backgrounds. We have developed a set of authoring tools for the content which are specifically designed to support collaborative authoring by multidisciplinary teams. Teams of authors develop specifications for interactive game scenes and lessons using Web-based tools. Automated tools analyze these specifications to detect errors and inconsistencies. The scene specifications are used to create interactive models of conversational agents, which are analyzed and tested for coverage. Recording scripts for voice actors are automatically extracted from the scene and lesson specifications. Finally, the scene and lesson descriptions are compiled into executable code that is loaded into the Unreal Tournament game engine to create the interactive learning system.

Two training systems have been developed using the TLTS: *Tactical Levantine Arabic* and *Tactical Iraqi*. As of May 2005 the Levantine Arabic version comprised a total of four interactive scenes and ten lessons. The Iraqi Arabic version encompasses seven training scenes, 1 test scene to evaluate learner proficiency at the end of the course, 22 lessons, and a post test. These support about 80 hours of learning time. The program has undergone iterative formative evaluations at USC, the US Military Academy, and the language training center at Ft Bragg, NC. Extensive beta testing of *Tactical Iraqi* with US military units was performed in the spring of 2005. Based upon these evaluations, *Tactical Iraqi* is being put into use in several military installations in the United States and Europe.

Our goal for *Tactical Iraqi* is to use the software to train US military personnel who are scheduled to be deployed on missions in Iraq. A new version of Pashto is being developed, and support for other languages and cultures is planned for the future. The training approach is not military-specific, and can be applied in any situation where learners must acquire enough knowledge of carry out specific tasks, e.g., book travel and accommodations, make an appointment for a business meeting, etc.

## 2. Illustrations

The following are some illustrations of *Tactical Levantine Arabic* in use. In the first image, the player (at near left) is greeting a man in the village (right). Behind him, far left, is the player's aide, who is monitoring the conversation and is able to offer advice or intervene if the player runs into difficulties. The second image shows the training interface, where the player can hear a phrase, attempt to repeat it, and receive feedback. The learner clicks on one of the speaker icons in the left column to hear the tutor speak a phrase. The learner then clicks on the microphone icon to speak the phrase and receive feedback.



## 3. Plan for Interaction with the Audience

The session will start with brief overview of the objectives of the project, and then give an in-depth demonstration of the system and its principal features. The participants will be

encouraged to use the software themselves to learn some spoken Arabic. We will discuss how effective the software is at meeting its design objectives, with implications for other AIED systems.

In the next part of the event we will look at some of the authoring tools and content used to create one of our training systems. We will examine content specifications for lessons and interactive scenes, and demonstrate some of the tools that we use to create the specifications. We will then demonstrate some of the tools that are used to test and validate the content specifications. We will also describe the automated production process that makes it possible to develop the large quantities of content needed for a Tactical Language course.

Next, we will look at the process of creating a training system using the authored lesson materials. We will show how XML specifications of content are translated into a form that can be executed in the TLTS engine.

Finally, we will look at some of the log files and data summaries that are created using the TLTS. The TLTS is an assessment tool as well as a training tool, and data collection and analysis plays an important role in all phases of the development process. These include speech recordings, interaction logs, survey data, and exams and quizzes. These provide process measures as well as outcome measures, and assessment of learner motivation and attitudes as well as acquired skills.

Here are some of the questions that will serve as the basis for discussion:

- The virtual tutor in Tactical Language attempts to emulate tutorial expertise, employing a combination of cognitive and motivational tactics. How successful is it in doing this? How might it be improved?
- What features of games are most important for promoting learning?
- How do game techniques integrate with motivational tutoring and other approaches?

#### **4. Acknowledgements**

This project is part of the DARWARS Training Superiority Program of the Defense Advanced Research Projects Agency. The authors wish to acknowledge the contributions of the members of the Tactical Language Team.

#### **References**

- [1] Beck, J. & Sison, J. (2004). Using knowledge tracing to measure student reading proficiencies. Proc. of ITS 2004. Berlin: Springer-Verlag.
- [2] Brown, P. & Levinson, S.C. (1987). Politeness: Some universals in language learning. New York: Cambridge University Press.
- [3] Johnson, W.L., Wu, S. & Nouhi, Y. (2004). Socially intelligent pronunciation feedback for second language learning. In Proc. of the ITS '04 Workshop on Social and Emotional Intelligence in Learning Environments.
- [4] Lepper, M.R., Woolverton, M., Mumme, D., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S.P. Lajoie and S.J. Derry eds., *Computers as cognitive tools*, 75-105. Hillsdale, NJ: Lawrence Erlbaum Associates.
- [5] Reeves, B. & Nass, C. (1996). *The Media Equation*. New York: Cambridge University Press.

# Rapid ITS Prototyping with Ontologies in WETAS

Brent Martin, Antonija Mitrovic and Pramudi Suraweera

*Intelligent Computer Tutoring Group, University of Canterbury,  
Private Bag 4800, Christchurch, New Zealand  
{brent,tanja,psu16}@cosc.canterbury.ac.nz*

**Abstract.** Participants will use a new authoring tool prototype to develop the domain model for a constraint-based ITS for search engine queries by producing an ontology. The users will develop this ontology diagrammatically. The system will then use it to generate the domain model, which can be imported into the WETAS ITS shell to produce a fully working ITS. Participants may also modify the generated domain model to improve feedback or correct errors. They will then run each other's tutors and discuss strengths and weaknesses of the approach.

## 1 Introduction

The WETAS system [1] supports the authoring of ITS by providing an engine that performs the main functions of a tutor (interface, account/security handling, student modeling, pedagogical functions). In particular, for text-based tutors the user only provides the domain model and problem set; WETAS does the rest. However, authoring the domain model is still hard. We are researching ways to make this easier by allowing the author to specify the domain model in the form of an ontology, from which WETAS derives the underlying rules of the domain. (In the case of WETAS, which implements constraint-based tutors, the domain model is represented as a set of constraints [2]).

Participants will use WETAS to develop a system for teaching internet users how to write search engine queries. This domain has been chosen because it is suitably small that participants might be expected to complete it in the time available. Participants will be scaffolded such that they only need to produce the ontology; functions normally performed by an author (writing the problem statements in particular) will have been completed by the event organizers. The participants will then be able to trial one another's tutors. The session will end with a discussion of their experiences.

## 2 Goals

The goals of the event are:

1. To introduce participants to the WETAS tool, and give them experience in developing constraint-based tutors using it, and;
2. To assess the feasibility of this approach to domain modeling

The goal (creating a domain model for search engine queries) will be broken into several subtasks so that all participants should be able to achieve a working tutor. These subtasks will be of increasingly complexity. The search engine language chosen is essentially a representation for logical expressions, with some additional keywords for search-specific conditions. We will structure the problem set such that the first few problems have simple expressions, requiring only a subset of the full domain of logical expressions. Later

problems will have more complex expressions, while the last few will have very complex expressions, necessitating the full language be modeled, and also the additional keywords. Additionally, some logical expressions are equivalent (by DeMorgans theorem, for example). The participant might first ignore this feature of the domain, and then later add it in.

### 3 Participants

The target group for this event is anyone interested in writing Intelligent Tutoring Systems. Skeptics are welcome. No particular skills are required: we are particularly interested in how easy/hard different types of users find this approach. Participants will be requested to fill in an anonymous questionnaire giving some details of their background. The organizers will provide help as required; whilst an objective of the study is to determine the viability of the approach, a secondary goal is for participants to experience writing constraint-based tutors. Ideally all participants will produce working tutors.

### 4 Description of the tool

The WETAS authoring tool produces constraint-based tutors. The domain model is represented by a set of modular “constraints” – rules that check that the student’s solution is in a valid state. If a constraint is violated feedback is given. WETAS has been trialled successfully for three consecutive years in a postgraduate Computer Science class learning about intelligent tutoring. In all cases almost all students succeeded in creating workable tutors. However, the quality of the tutoring systems has been variable, with the domain model being the source of most problems. Two major difficulties have been identified: the students do not capture the states requiring modelling correctly, and they fail to properly encode them into constraints.

Until now, the domain model has been written by hand-coding the constraints using a text editor. Although the language used is fairly straightforward (a pattern-matching language based on LISP), the encoding of the constraints can get quite complicated when complex structures are involved in the domain. Further, the number of constraints can get quite large – our largest tutor contains nearly 1000 constraints. Keeping track of the constraints is difficult, with oversights being inevitable.

However, we contend that much of this work is in fact redundant. First, many constraints have similar “patterns” that check for relevance/correctness. Second, many of the feedback messages attached to the constraints are similar, differing only by some small detail, or by the *object* of the message (“*you are missing a table*” versus “*you are missing an attribute*”). We analysed one of our tutors (SQL-TUTOR) and identified a set of “meta-constraints” – constraints that when applied to every object in the domain model (and relationships between them) cover all of the constraints in the original set. This reduces the authoring task to that of specifying all of the domain objects and the relationships between them.

Ontologies are a useful tool for describing objects and relationships [3]. They enable the problem space (describing the domain) to be broken down hierarchically into manageable sub-parts, thus allowing the model to be viewed at varying levels of abstraction. We have developed a tool that allows the user to specify the domain diagrammatically as an ontology. Nodes in the diagram specify the objects of the domain. These contain slots for the object name, how it is identified in a solution, and any constraints upon this object (such as a maximum cardinality). Arcs represent relationships, specifying constraints upon



combinations of objects, including whether they are alternatives or sequences of objects that must all appear together, for example. The system then maps the “meta-constraints” onto the ontology, producing a set of constraints for the domain. This process can occur at any time; the ontology does not need to be complete, although the quality of the resulting constraint set will of course be dependent on the quality of the ontology.

## 5 Event structure

The structure of the event will be as follows:

- Presentation by the organizers outlining the purpose of the event and describing and demonstrating WETAS and some existing tutors written using it;
- Participants are walked through logging into the tool and starting their authoring session
- The organizers present a brief tutorial on how to use the tool, and describe the task domain (search engine queries). Material needed (i.e. a description of the search engine language being used) is disseminated;
- The participants work in the tool producing an ontology for the search engine language. They are able to generate a working tutor from their ontology at any time.
- After a fixed time period participants are invited to swap tutors with their neighboring participant and try using them (as students) to gauge effectiveness, and discuss their experiences
- Towards the end of the session the organizers will lead a discussion. Topics covered will include:
  - How far did everyone get?
  - How intuitive (or not) did they find the general approach?
  - Did they think the approach suited the domain? What domain types might it *not* suit?
  - How easy/hard to use was the tool?
  - What were their impressions of the quality of the tutor produced?
  - Did trying someone else’s tutor give them any insight into how they might improve their own?
  - What improvements could be made?
  - Did they think they might be tempted to build tutors using this approach in the future if the tool was made available (e.g. over the internet?)

Participants will use individual PCs to interact with the ontology editor and WETAS systems.

## 6 Related contributions to AIED

- Suraweera, P., Mitrovic, A. and Martin B. A Knowledge Acquisition System for Constraint-based Intelligent Tutoring Systems (full paper).

## References

- [1] Martin, B. and Mitrovic, A. *WETAS: A Web-Based Authoring System for Constraint-Based ITS*. in *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. 2002. Malaga: Springer. p. 543-546.
- [2] Mitrovic, A., Mayo, M., Suraweera, P., and Martin, B. *Constraint-Based Tutors: A Success Story*. in *Fourteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*. 2001. Budapest, Hungary: Springer. p. 931-940.
- [3] Mitrovic, A., Suraweera, P., and Martin, B. *The role of domain ontology in knowledge acquisition for ITS*. in *Seventh international conference on Intelligent Tutoring Systems*. 2004. Maceio, Brazil: Springer. p. 207-216.

# OntoAIMS: Semantically-aware and User-adaptive Learning Content Management

Lora AROYO<sup>1</sup>, Vania DIMITROVA<sup>2</sup>

<sup>1</sup>*Faculty of Mathematics and Computer Science  
Eindhoven University of Technology, The Netherlands*

[l.m.aroyo@tue.nl](mailto:l.m.aroyo@tue.nl)

<sup>2</sup>*School of Computing  
University of Leeds, LS2 9NA, Leeds UK*  
[vania@comp.leeds.ac.uk](mailto:vania@comp.leeds.ac.uk)

There is a strong argument that the user's perspective has to be taken into account to ensure that the Semantic Web can improve the way computers and people work together. Personalization technologies for the Semantic Web are needed to deal with user diversity. User-adaptive systems for the Semantic Web need approaches for user modelling that work, unlike traditional approaches, under the open-world assumption. A novel framework for eliciting a user's conceptualization based on an ontology-driven dialog is presented here.

Our goal is to demonstrate how we can construct and maintain an *enhanced user model* that integrates different user perspectives, such as knowledge, personal preferences, interests, browsing patterns, cognitive and physical state. We show how we utilize this user model for web-based personalized content management and user-adaptive information retrieval. We illustrate this in an RDF/OWL-based adaptive semantic-web system for personalized learning content management, called OntoAIMS. The user model in OntoAIMS is based on the collection, interpretation and validation of user data from diverse sources, such as user-defined preferences, diagnostic dialog with the user, and monitoring the user's behaviour with the system (e.g. links followed, resources opened, and searches performed). The diagnostic dialog with the user introduces a novel approach based on a series of dialog games where a user and the system discuss and construct together a model of the user's conceptualization of the domain [1]. The approach is illustrated in OWL-OLM – an OWL-based version of the STyLe-OLM [2] tool for open user modelling, which was originally based on conceptual graphs.

During the Interactive Event we will present the OntoAIMS and the OWL-OLM to the audience in the context of a learning scenario dealing with the cold start problem and tailoring the system's behaviour to the needs of each individual learner. The users will be asked to perform several tasks, related to choosing a course task to work on and searching for relevant resources within the context of a course task. At the end of the session the users will be asked to fill in a questionnaire related to their satisfaction and suggestions for system improvement.

## References

- [1] Denaux, R., Dimitrova, V., Aroyo, L.: Integrating open user modeling and learning content management for the semantic web. In: International Conference on User Modeling, UM05.
- [2] V. Dimitrova. STyLe-OLM: Interactive open learner modelling. Int. Journal of Artificial Intelligence in Education, 13(1):35–78, 2003.



# Exploring meta-cognitive tutoring by the Help Tutor: An Interactive Event

*Vincent Aleven, Bruce McLaren, Ido Roll, and Ken Koedinger*

Human-Computer Interaction Institute  
Carnegie Mellon University  
5000 Forbes Ave, Pittsburgh PA 15213, USA  
aleven@cs.cmu.edu, bmclaren@cs.cmu.edu, iroll@andrew.cmu.edu,  
koedinger@cmu.edu  
+1 (412) 268-5475

**Abstract.** During this Interactive Event in which the participants will do some task-driven exploration of the *Help Tutor*, a metacognitive tutor agent we have developed to help students become better help seekers as they work with a Cognitive Tutor. The *Help Tutor* is based on a cognitive model of students' desired help-seeking processes and is implemented as 74 production rules that represent both productive and unproductive help-seeking behavior (Aleven et al., 2004; in press-b). We designed an architecture in which the Help Tutor and a Cognitive Tutor function as independent agents to facilitate re-use of the Help Tutor. A small, preliminary study showed that students improved their help-seeking behavior significantly while working with the Help Tutor, and we are planning a larger classroom study to test the impact of the Help Tutor on both learning and help-seeking behavior. In this Interactive Event, we will give the participants an opportunity to use the Help Tutor and to experience directly the metacognitive feedback it provides.

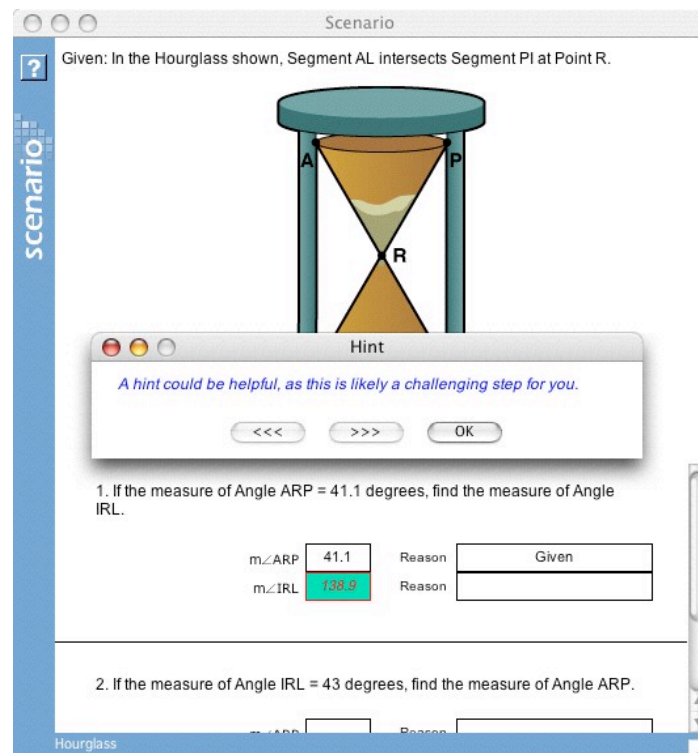
## General description of the Interactive Event

The participants of this Interactive Event will have an opportunity to experience and observe the Help Tutor, a metacognitive tutor agent designed to help students become better help seekers. The Interactive Event will be a mixture of lecture, demonstration, and hands-on use of the Help Tutor. During the hands-on part, the participants will be given an inquiry task, aimed at “discovering” the Help Tutor’s underlying model of help seeking, as a way to stimulate productive discussion.

Our work on the Help Tutor draws from a variety of research focused on metacognition (e.g., Bielaczyc, Pirolli, & Brown, 1995; Palincsar & Brown, 1984; White & Frederiksen, 1998). We are particularly interested in how we can teach meta-cognitive skills, in addition to helping students at the domain level. While a number of efforts have focused on supporting metacognition in intelligent tutoring systems (e.g., Aleven & Koedinger, 2002; Bunt, Conati, & Muldner, 2004; Conati & VanLehn, 2000; Gama, 2004; Luckin & Hammerton, 2002), ITS researchers are only beginning to explore the *value* of supporting metacognition in ITSs.

Our particular interest is in help seeking and has resulted in the development of the Help Tutor, a tutor agent that uses a cognitive model of adaptive help-seeking behavior to provide feedback on students’ help-seeking behavior (Aleven et al., 2004; in press-a; in press-b; Roll et al., in press). The model stipulates, for example, that students work deliberately, that they use help on unfamiliar steps or when they make repeated errors on a step, and that they read tutor messages carefully. The Help Tutor is meant to be used in conjunction with a Cognitive Tutor that teaches “at the domain level,” for example, algebra or geometry. So far, we have integrated it with the Geometry Cognitive Tutor, a tutor for high-school geometry problem solving. This tutor monitors students’ step-by-step problem solutions using a cognitive model of student problem solving. It provides feedback and, at the student’s request, context-sensitive hints at the domain level, explaining how a certain geometry theorem or definition

can be applied to solve the step that the student is working on. The tutor also provides a second form of help, a searchable on-line Glossary with detailed information about the relevant geometry theorems and definitions, which students can browse freely. In the integrated Geometry + Help Tutor version, students receive not only assistance with respect to their attempts at solving geometry problems, the Help Tutor also gives feedback on how they use the tutor's help facilities. For instance, in Figure 1, the Help Tutor indicates to the student that asking for a hint would be appropriate and useful. It makes such decisions based on the Cognitive Tutor's assessment of that student's competence at the skill involved in the current step and based on the number of errors made so far on the given step.



**Figure 1:** A suggestion provided by the Help Tutor while a student works on a geometry problem

The hypothesis underlying this research is that the Help Tutor will help students to become better help seekers, that they will learn better as they are using the Help Tutor, but also that they will become better future learners (i.e., that they will continue to learn better even after the Help Tutor is removed). If our planned evaluation studies confirm this hypothesis, that would also provide stronger causal evidence for the influence of help seeking on learning than other work has established, within the realm of computer-based instruction.

### Goal of the Interactive Event

The participants will get first-hand experience using a meta-cognitive tutor designed to support help-seeking behavior in students. They will have an opportunity to discuss issues related to how intelligent tutoring systems can best support and help students develop meta-cognitive abilities such as help seeking.

### Target group of the Interactive Event

The target audience of the event includes:

- Researchers who are studying meta-cognition.

- Learning science researchers interested in how help-seeking behavior can be supported in educational technology.
- Educators familiar with Cognitive Tutors who are interested in new developments to improve the effectiveness of the tutors.
- Intelligent tutor developers interested in modular architecture and the concept of adding a “tutor agent” to an existing architecture.

### **Prerequisite knowledge and skills needed to participate in the event**

No prerequisite skills are required other than a basic amount of computer literacy and an interest in the development of educational software.

### **Organization of the event**

The event will have four main sections

1. Introduction (15 mins)
2. Demonstrate the use of the Help Tutor (5 mins)
3. Hands-on use of the Help Tutor (30 mins)
4. Discussion (40 mins)

During the *introduction*, we will provide background information to help participants understand the context of the research. Specifically, we will briefly present data about students’ ineffective help-seeking behavior and the extent to which it appears to lead to reduced learning gains. We will then briefly discuss with the participants what good help seeking with a Cognitive Tutor might be like. However, we will not yet present the model of help seeking that underlies the Help Tutor. Rather, during the hands-on session of the event, the participants will investigate how the Help Tutor reacts to students’ to “discover” its underlying model of help seeking.

During the *demonstration* of the Help Tutor, we will first show the basic help functions of the geometry Cognitive Tutor, with the Help Tutor turned off. We will then demonstrate some scenarios with the Help Tutor turned on to illustrate the feedback that the Help Tutor gives.

Next, during the *hands-on part* of the event, the participants will first work through a problem or two with the Cognitive Tutor with the Help Tutor turned off, to get used to the tutor interface, its help functions, etc. They will then use the tutor for a problem or two with the Help Tutor turned on, so as to get a first impression of the Help Tutor feedback. Then they will be given an inquiry task, namely, to investigate the model of adaptive help seeking that underlies the Help Tutors. (We may assign each participant or group of participants the task of investigating one particular aspect of the Help Tutor, e.g., how it reacts to help avoidance, or how it reacts to students venturing repeated guesses.) They can carry out this task by coming up with a set of scenarios that represent a range of (presumably) productive and unproductive help-seeking behavior and try them out on the tutor. The point of doing this is not so much that the participants get to know the Help Tutor’s underlying model in great detail but rather that they spend some time thinking about what good help-seeking behavior might be like and how students are likely to deviate from that behavior. Of course, we will be very interested in hearing whether they think the Help Tutor reacts adequately to different kinds of unproductive help-seeking behavior. We will also be very interested in any suggestions the participants have about how the Help Tutor could behave differently.

During the *discussion* section, we would like to ask each participant group to “report” back – e.g., how they expected the Help Tutor to react, and how it did or did not conform to their expectations. In the course of this discussion, we will “reveal” the model of adaptive help seeking that underlies this tutor. The point is not, of course, to hold up this model as “the correct answer” to the exercise that was assigned. Rather, at that point the participants would be optimally prepared to give their critical comments on the model.

## Discussion plan

The discussion session will touch upon the topics that participants bring up as well as on any or all of the following issues:

- What is good help seeking?
- What might cause ineffective help-seeking behavior?
- How was the model of help-seeking behavior used in the Help Tutor created and how was it validated?
- To what extent is the use of the Help Tutor likely to lead to more adaptive help-seeking behavior by students in other settings such as classrooms?
- How should ideal help seeking be taught? What methods should be used in conjunction with the Help Tutor feedback on Help Seeking bugs?
- Does the Help Tutor intervene with appropriate frequency, that is, enough to make a difference but not so much as to be considered a “nag” by the students?
- What would it take to use the Help Tutor with other intelligent tutoring systems?
- Should the Help Tutor dynamically alter its behavior based on the history of individual students?

## References

- Aleven, V., & Koedinger, K. R. (2002). An Effective Meta-cognitive Strategy: Learning by Doing and Explaining with a Computer-Based Cognitive Tutor. *Cognitive Science*, 26(2), 147-179.
- Aleven, V., McLaren, B. M., & Koedinger, K. R. (in press-a). Towards Computer-Based Tutoring of Help-Seeking Skills. In S. Karabenick & R. Newman (Eds.), *Help Seeking in Academic Settings: Goals, Groups, and Contexts*. Mahwah, NJ: Erlbaum.
- Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2004). Toward tutoring help seeking: Applying cognitive modeling to meta-cognitive skills. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.), *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 227-239). Berlin: Springer Verlag.
- Aleven, V., Roll, I., McLaren, B. M., Ryu, E. J., & Koedinger, K. R. (in press-b). An architecture to combine meta-cognitive and cognitive tutoring: Pilot testing the Help Tutor. In *Proceedings of the 12<sup>th</sup> International Conference on Artificial Intelligence in Education, AIED 2005*.
- Bielaczyc, K., Pirolli, P. L., & Brown, A. L. (1995). Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem Solving. *Cognition and Instruction*, 13, 221-252.
- Bunt A., Conati C & Muldner K. (2004). Scaffolding self-explanation to improve learning in exploratory learning environments. In *Proceedings 7th Intern. Conf. on Intelligent Tutoring Systems, ITS 2004*. Berlin: Springer.
- Conati C. & VanLehn K. (2000). Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education*, 11, 398-415.
- Gama, C. (2004). Metacognition in Interactive Learning Environments: The Reflection Assistant Model. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.), *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004*. Berlin: Springer Verlag. Berlin: Springer.
- Luckin, R., & Hammerton, L. (2002). Getting to know me: Helping learners understand their own learning needs through meta-cognitive scaffolding. In *Proceedings of Sixth International Conference on Intelligent Tutoring Systems, ITS 2002* (pp. 759-771). Berlin: Springer.
- Palincsar, A. S., & Brown, A. L. (1984). Reciprocal Teaching of Comprehension-Fostering and Comprehension Monitoring Activities. *Cognition and Instruction*, 1, 117-175.
- Roll, I., Baker, R.S., Aleven, V., McLaren, B., & Koedinger, K. (in press). Modeling Students' Metacognitive Errors in Two Intelligent Tutoring Systems. In *Proceedings of the 10th International Conference on User Modeling (UM'2005)*.
- White, B. & Frederiksen, J. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction*, 16(1), 3-117.



# Human Centred Design for interactive educational software

*Rosemary LUCKIN, Geraldine FITZPATRICK,  
IDEAS Lab, Interact Lab  
Human Centred Technology Group  
Department of Informatics, University of Sussex  
Brighton, BN1 9QH, UK  
Human Centred Technology Group, Department of Informatics, University of Sussex,  
Brighton, BN1 9QH, UK  
<http://www.informatics.sussex.ac.uk/research/hct/>  
<http://www.informatics.sussex.ac.uk/ideas/>  
<http://www.informatics.sussex.ac.uk/interact/>*

## 1 Introduction and Background

At this interactive event members of the Human Centred Technology group at Sussex University will offer participants the chance to try out some of the educational software and hardware we are involved in developing. This will include hands-on interactions with prototypes from purely paper based though semi-functional to fully specified applications and the chance to explore the data and techniques we use for evaluation purposes including video, audio, photos and log files. Through this event we hope to demonstrate the benefits of the human centred design methodology and allow people to experience its complexities and delights.

The Human Centred Technology (HCT) Research Group at Sussex is an internationally renowned research centre that develops interactive learning environments with a focus on understanding how people learn when using technology across multiple contexts and artifacts. We produce educational software as research tools, evaluate existing technology and produce guidelines for the design and use of digital technology within educational contexts. We adopt a learner centred iterative design approach to building interactions between learners and technology and conduct our work in collaboration with the users and learners for whom we are designing. This process ensures that the user's needs drive the development of the technology

## 2 General Description of the Interactive Event

The goal of this event is to unpack the human centred design approach by inviting people to interact with a variety of pieces of educational technology at different stages in their development. These stages range from paper prototype to functional application. In parallel with the presentation of each stage of technology development participants will be able to explore the data collected from the associated user engagement activities.

The target group for the event would be any researcher, developer or practitioner who wants to understand more about how they might use some aspects of this approach, such as the development of low-tech prototypes or a particular analytical technique.

### *The Event*

The event will be split into 6 project experience zones, each of which will illustrate how a particular project has developed, its technology including opportunities to interact with different prototypes and explore the user data collected at each stage of evaluation. Each zone will make a brief presentation before allowing participants to visit the zones in turn to find out more. We would repeat these presentations at regular intervals to enable participants to visit multiple zones. Each zone would be able to accommodate up to 6 participants at any one time.

### *Zone Descriptions*

#### *Homework*

The Homework project is developing an exemplar system for the delivery of adaptive, interactive numeracy and literacy education for children aged 5 to 7 years at home and in the classroom using a combination of interactive whiteboard, tablet PC and wireless technology. In order to operate effectively in the multiple contexts of home, school and in between we have conducted a series of user studies employing different interpretations and manifestations of the homework 'system'. These have included scenarios and mock-ups as well as semi-functional but as yet non-adaptive software prototypes. These user engagement activities have required a variety of data collection and analysis methodologies including home diaries, interviews, logging and observational data. In the homework zone participants will be able to try out some of the activities on the tablet PCs, work through the scenarios and explore and compare their own activities with those of the users as reflected in the annotated logs and diaries we have collated.

#### *Riddles*

The Riddles project is designing and developing educational software that focuses on improving children's reading comprehension through their collaborative engagement with activities to increase Language Awareness. In this zone, participants will be invited to explore our prototypes in Lego, powerpoint and paper/cardboard formats, as well as engage with semi-functional software. Data gathered from this project includes coded collaborative interactions and video footage of children interacting with our prototypes throughout all stages of their development, which have all contributed to the iterative design cycle. Participants will be invited to reflect upon the issues that we have found important and to discuss how these relate to their own research.

#### *Adventure Author – in collaboration with Judy Robertson (Glasgow Caledonian University)*

Adventure Author is a joint collaboration between Glasgow Caledonian University and the IDEAS Lab. The Adventure Author software enables 10-14 year old children to create interactive stories in a 3D computer game medium in which their peers can participate. Adventure Author provides an environment in which to create and participate in interactive stories of commercial quality and, as such, has been found to be highly motivating in preliminary evaluations. In addition, the adventures created are built around interaction with characters through dialogue, and Adventure Author therefore provides support for a number of key aspects of narrative development. The Adventure Author design process has involved a number of stages, including user centred design sessions with children, and an exploration the feasibility of the initial concept using existing commercial games. In the

Adventure Author zone, participants will be able to follow the design process used, interact with Adventure Author to create stories, and play stories created by other participants.

### *EScience*

This project has been exploring how eScience might be used to provide hands-on collaborative learning experiences for school children involving real university scientific research, data, devices and scientists. We ran several sessions with teachers and learners relating to two real e-Science projects, remote sensing on an Antarctic lake and mobile urban pollution (CO) sensing and visualization. The intention of these sessions was to simulate the kinds of activities that might be possible with e-Science and to elicit teacher and learner reactions and suggestions as well as to gather data on requirements for scaling up and improving. The eScience zone will enable participants to re-discover their exploring skills and re-live some of the activities children engaged in; web-based exploration of the remote sensing device, live collaboration with remote scientists in the Antarctic and elsewhere, mobile CO/GPS sensing and visualization and reflection on what school might be like with real access to e-Science. Visitors to the e-Science zone will also be able to look at some of the data we collected, the children's data-review software, and research data analysis schemes under development.

### *Ambient Wood - Hilary Smith*

In Ambient Wood, we developed a novel interactive learning experience for children aged 11-12 years old that took them out of the classroom setting to a woodland habitat that was augmented with a wireless network and various devices and sensor-based technologies. This was an exploratory project to understand if and how augmenting the physical environment in this way could promote reflective skills in the context of learning about scientific enquiry. The children's task was to explore the local habitat and they were supported in this through the provision of various digital visualisations and sounds, closely coupled with their physical explorations. For example, they could use a probe tool to measure moisture and light, collect cards of plants and animals sent to their PDA by a remote facilitator after they had described what they were seeing, listen to sounds that were triggered by their explorations suggesting some 'invisible' process in the wood, and so on. Visitors to the Ambient Wood zone will be able to interact with some of the devices used as well as look at research video recordings and other data collected in the course of the sessions. We will be able to discuss the key findings arising from the analysis and also the practical challenges involved in staging such an 'in the wild' learning experience and integrating multiple streams of digital, video and audio data for analysis.

### *Exploring Interfaces and Interactions – Lucidna Kerawalla, Rosemary Luckin*

Visitors to this zone will have the chance to try out several of the different interfaces we have been using in recent studies. These will include an augmented reality (AR) 'virtual mirror' mixed-reality interface developed using the ARToolkit software, a SmartBoard digital whiteboard system (including projector), a web camera (positioned near or at the whiteboard/screen projection plane), and virtual 3D science content represented as 2D graphics on cards. In a recent collaboration with the BBC creative research and development team young learners were able to interact with a virtual 3D model of the earth, sun and moon. Visitors will be able to try out the interface for themselves, discuss its implications for classroom learning and consider the data and results arising from our empirical studies.



# Interactive Demonstration of a Generic Tool to Browse Tutor-Student Interactions

Jack Mostow, Joseph Beck, Hao Cen, Evandro Gouvea, and Cecily Heiner  
*Project LISTEN* ([www.cs.cmu.edu/~listen](http://www.cs.cmu.edu/~listen), 412-268-1330), Carnegie Mellon University  
 RI-NSH 4213, 5000 Forbes Avenue, Pittsburgh, PA. USA 15213-3890

**Abstract.** Project LISTEN's Session Browser is a generic tool to browse a database of students' interactions with an automated tutor. Using databases logged by Project LISTEN's Reading Tutor, we illustrate how to specify phenomena to investigate, explore events and the context where they occurred, dynamically drill down and adjust which details to display, and summarize events in human-understandable form. The tool should apply to MySQL databases from other tutors as well.

## 1. Introduction

A basic question in mining data from an intelligent tutoring system is, "What happened when...?" We demonstrate a tool to help answer such questions. It lets users specify phenomena, find where they occur in the data, and browse them in human-understandable form. The tool applies to MySQL databases whose representation of tutorial events includes student, computer, start time, and end time [1]. It automatically computes and displays the temporal hierarchy implicit in this representation, as explained in [2]. Here we illustrate the use of this tool to mine data from Project LISTEN's Reading Tutor [3].

## 2. Select events to explore.

As an example, we focus on a particular student behavior: clicking *Back* out of stories. The Reading Tutor has *Go* and *Back* buttons to navigate to the next or previous sentence in a story. We had previously observed that students sometimes backed out of a story by clicking *Back* repeatedly even after they had invested considerable time in the story. We are interested in understanding what might precipitate this undesirable behavior.

<pre>select * from story_encounter where Exit_through = 'user_goes_back' and (unix_timestamp(End_time) - unix_timestamp(Start_time) &gt; 60) order by rand() limit 10</pre>							<b>Table: STORY_ENCOUNTER</b> <b>Start_Time</b> 2004-12-02 08:30:31.609 <b>End_Time</b> 2004-12-02 08:34:17.156 <b>Machine_Name</b> LISTEN01-321-04 <b>Session_Start_Time</b> 2004-12-02 08:30:28.0 <b>Exit_Through</b> user_goes_back <b>Story_Directory</b> New Story Fall 2003 - Earthworms <b>User_ID</b> mTJ5-7-1995-12-03 <b>Student_Level</b> A <b>Initiative</b> student_initiative
Start_Time	End_Time	Machine_Na...	Session_St...	Exit_Through	Stor...	Us	
2004-10-20 ...	2004-10-20 ...	LISTEN01-30...	2004-10-20 ...	user_goes_...	New...	fKD5-	
2004-12-02 ...	2004-12-02 ...	LISTEN01-32...	2004-12-02 ...	user_goes_...	New...	mTJ5-	
2005-01-07 ...	2005-01-07 ...	LISTEN01-27...	2005-01-07 ...	user_goes_...	New...	mMB4	
2004-12-08 ...	2004-12-08 ...	LISTEN01-27...	2004-12-08 ...	user_goes_...	CMU...	fTS6-	
2005-01-07 ...	2005-01-07 ...	LISTEN01-30...	2005-01-07 ...	user_goes_...	CLO...	mDY5	
2004-12-02 ...	2004-12-02 ...	LISTEN01-30...	2004-12-02 ...	user_goes_...	New...	mDF6	
2004-11-10 ...	2004-11-10 ...	LISTEN01-33...	2004-11-10 ...	user_goes_...	New...	fEF7-	
2005-01-05 ...	2005-01-05 ...	LISTEN01-27...	2005-01-05 ...	user_goes_...	New...	fTVW7-	
2004-11-18 ...	2004-11-18 ...	LISTEN01-28...	2004-11-18 ...	user_goes_...	CMU...	fDA7-	
2005-01-06 ...	2005-01-06 ...	LISTEN01-33...	2005-01-06 ...	user_goes_...	New...	fLR7-	

Figure 1: a. Query and its resulting table of events. b. Attribute-value list for selected event.

The first step is to find examples of the phenomenon of interest. The query in Figure 1a finds a random sample of 10 stories that students backed out of after spending more than a minute in the story. The resulting table has a column for each field of “story\_encounter” in the database, including the start and end times of the story, the name of the computer that recorded it, when the session started, how the user exited the story (by finishing it, backing out, etc.), the name of the story, the user ID of the student, and so on. When the user selects a record in this table, the tool lists its field names and values as shown in Figure 1b. However, this information supports only limited understanding of the event, because it lacks context. As a companion paper [2] explains, the tool computes the context of the event as its temporally enclosing events, in this case the session where the story was read, and the student, whom the tool treats as an infinite time interval when it computes context.

### 3. Dynamically drill down and adjust which details to include.

How can we generate a dynamic, adjustable-detail view of hierarchical structure in a human-understandable, easily controllable form? We adapted a standard widget for expandable trees. Given a target event, the tool at first displays only its context, i.e., its direct ancestors, omitting other students, sessions, and stories. To see the offspring of the selected story “Earthworms Have An Important Job” or any other event, the user expands it by clicking on the “+” icon to its left. The folder icon marks events not yet fully expanded. Collapsing and re-expanding a partially expanded event reveals its other offspring. Figure 2 shows the result of expanding some details of the event, in particular the sentence encounters preceding the end of the story encounter, back to where the student started backing out of the story, as the Student\_Click “user\_goes\_back” events indicate.

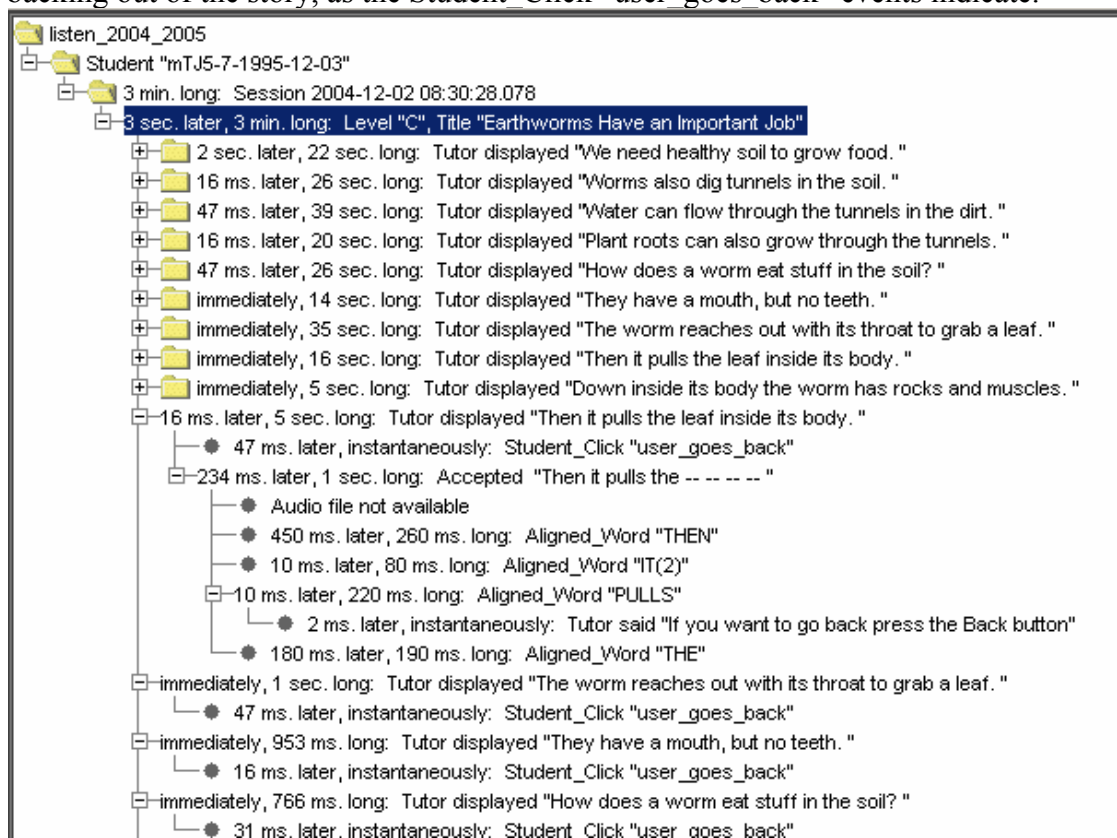


Figure 2: Hierarchical context and partially expanded details of a selected event

Expanding these details revealed a surprise: the Reading Tutor said to click *Back*. The student had clicked above the sentence. This event might mean the student wanted to return to the previous sentence – or that he was trying to click on a word for help but

missed the target. Due to this ambiguity, the Reading Tutor did not respond to “user\_clicks\_above\_sentence” by backing up, but just by saying “If you want to go back, press the Back button.” This example suggested the novel hypothesis that the Reading Tutor itself might unintentionally be prompting students to back out of stories!

Besides drilling down as above, the user can specify more globally which types of events to display. The “Pick tables” tab includes a menu that shows which database is currently selected. For instance, “listen\_2004\_2005” has data for the 2004-05 school year. We assume the database has a different table for each type of event. A checkbox for each table in the currently selected database specifies whether to include that type of event in the event tree. For example, turning on “audio\_output” shows speech output by the Reading Tutor, such as “if you want to go back press the Back button.”

The checkboxes do not distinguish among events of the same type. For instance, a user might want the event tree to include the tutor’s spoken tutorial assistance but not its backchannelling (e.g., “mmm”). User-programmable filters would allow such finer-grained distinctions, but be harder than using check boxes to specify which event types to include.

What steps might subsequent exploration pursue, with what support by the tool?

- Continue browsing this case to try to identify other possible reasons for backing out.
- Check other instances of backing out (by clicking on other events from the table in Figure 1a) to see if the Reading Tutor suggested it.
- Retrieve cases of the same Reading Tutor prompt by formulating a suitable query to enter in the query box (see Figure 1a) to see if the student backed out then as well.
- Develop a query to count how often students back out with vs. without such a prompt. This step constitutes a more quantitative phase of data mining, but the tool lets us inspect 10 randomly chosen cases to check if the query treats them correctly.

#### 4. Summarize events in a human-understandable form.

We have already described the event trees we use to convey the hierarchical structure of tutorial interaction. But how do we summarize individual events?

Temporal properties are common to all events, so we treat them uniformly. An event’s absolute start and end times seldom matter except for time of day or time of year effects. Therefore we display them only in the event’s attribute-value list, and for a session.

In contrast, the duration of an event is a simple but informative universal measure. For example, the fact that most of the sentence encounters before the student started backing out of the story lasted 14-39 seconds indicates a slow reader. The duration of an event is simply its end time minus its start time.

The hiatus between two events is informative because it reflects user effort, hesitation, confusion, or inactivity. For example, the fact that the hiatuses before *Back* clicks were less than 100 milliseconds long suggests that the student may have been clicking repeatedly as fast as possible. Consider a parent event A with offspring B and C: [A starts ... [B starts ... B ends] ... [C starts ... C ends] ... A ends]. The hiatus between a parent A and its first offspring B is the start time of B minus the start time of A. The hiatus between successive sibling events B and C is the start time of C minus the end time of B.

Precise times seldom matter for a duration or hiatus, so for readability and brevity, we display only the largest non-zero units (days, hours, minutes, seconds, milliseconds).

The complete attribute-value list for an event occupies considerable screen space, and is displayed only for the currently selected event. In contrast, the tool displays all the one-line summaries for an event tree at once. What information should such summaries include? How should it be displayed? How should it be computed?

The answers depend on the type of event. We observed that although the Reading Tutor’s database schema has evolved over time, the meaning of table names is nevertheless

consistent across successive versions and between databases created by different members of Project LISTEN. Therefore we wrote one method for each table to translate a record from that table into a one-line string that includes whatever we think is most informative. Ideally these methods are simple enough for users (educational data miners) to modify to suit their own preferences. The default string for a table without such a method is just the name of the table, e.g., “Session” or “Story\_encounter.” Most methods just display one or more fields of the record for the event. For example, the method for a session just shows its start time. Some methods incorporate information from other tables. For example, the method for a story encounter retrieves its title from a separate table. Special-purpose code adds a node the user can click to play back a recorded utterance, or displays “Audio not available” if its audio file has not yet been archived (as in the case of the December 2004 example shown here).

## 5. Evaluation

Relevant criteria for evaluating this work include implementation cost, efficiency, generality, usability, and utility. Implementation cost was only several person-weeks for a tutor-specific prototype, and about the same for its generic, time-interval-based successor.

Using ordinary PCs for the database server and the session browser to explore databases for hundreds of students, thousands of hours of interaction, and millions of words, the operations reported here usually update the display with no perceptible lag, though a complex query to find a specified set of events may take several seconds or more.

Structural evidence of generality includes the tool’s predominantly tutor-independent design, reflected in the code’s brevity and its scarcity of references to specific tables or fields of the database. Empirical evidence of generality includes successful use of the tool with databases from different years’ versions of the Reading Tutor. We have not as yet tested it on databases from other groups. Typically ITSs still log to files, not databases.

It is early to evaluate usability or utility because the tool is still so new. We have not conducted formal usability tests on its initial target users, namely Project LISTEN researchers engaged in educational data mining. However, we can claim a ten- or hundred-fold reduction in keystrokes compared to obtaining the same information by querying the database directly. For example, clicking on an item in the event list displays its context as a chain of ancestor events. Identifying these ancestors by querying the database directly would require querying a separate table for each ancestor.

**Acknowledgements:** This work was supported in part by the National Science Foundation under ITR/IERI Grant No. REC-0326153. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government. We thank the educators and students who generated our data, and our colleagues who contributed to developing the Session Browser.

## References (see [www.cs.cmu.edu/~listen](http://www.cs.cmu.edu/~listen))

1. Mostow, J., J. Beck, R. Chalasani, A. Cuneo, and P. Jia. Viewing and Analyzing Multimodal Human-computer Tutorial Dialogue: A Database Approach. *Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces (ICMI 2002)*, 129-134. 2002. Pittsburgh, PA: IEEE.
2. Mostow, J., J. Beck, A. Cuneo, E. Gouvea, and C. Heiner. A Generic Tool to Browse Tutor-Student Interactions: Time Will Tell! *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)* 2005. Amsterdam.
3. Mostow, J., G. Aist, P. Burkhead, A. Corbett, A. Cuneo, S. Eitelman, C. Huang, B. Junker, M.B. Sklar, and B. Tobin. Evaluation of an automated Reading Tutor that listens: Comparison to human tutoring and classroom instruction. *Journal of Educational Computing Research*, 2003. 29(1): p. 61-117.



# A tutoring system to practice theorem proving in Fitch

Sicco Verwer, Mathijs de Weerd, Jonne Zutt

*Delft University of Technology*

*{s.e.verwer, m.m.deweerd, j.zutt}@st.ewi.tudelft.nl*

*015-2784516*

## Abstract

In this interactive event we demonstrate a web-based software tool to teach theorem proving in propositional logic, called Bop. This tool is a proof editor in the Fitch proof system that can give hints, proofsteps, or even complete proofs to the student.

**Keywords:** Theorem proving, tutoring system

## 1 Introduction

This interactive event is intended to demonstrate a software tool to assist bachelor students in learning to prove theorems in propositional logic, called Bop. In this document we first describe the tool Bop. Then we give the details of the event itself.

## 2 Description of Bop

This interactive event demonstrates a software tool, called Bop, which assists bachelor students in learning to prove theorems in propositional logic. This tool is a proof editor in the Fitch proof system [1]. The Fitch proof system uses an intuitive proof notation in the form of diagrams. This notation and the simple rules of inference make the Fitch system an ideal system for learning logic. Bop consists of a web-based graphical user interface (the client) and a theorem prover for propositional logic (the server). This setup allows students to learn how to construct proofs from wherever they have an internet connection. A second-year course on logic at the Delft University of Technology makes use of this tool [3]. The basic idea of the Bop program is as follows:

- A student would like to (learn to) prove a theorem and enters this theorem in the web-client program.
- The student first tries to construct the proof himself using the interface to edit the proof.
- The program automatically detects proven steps and notifies the student when he has completed the proof.

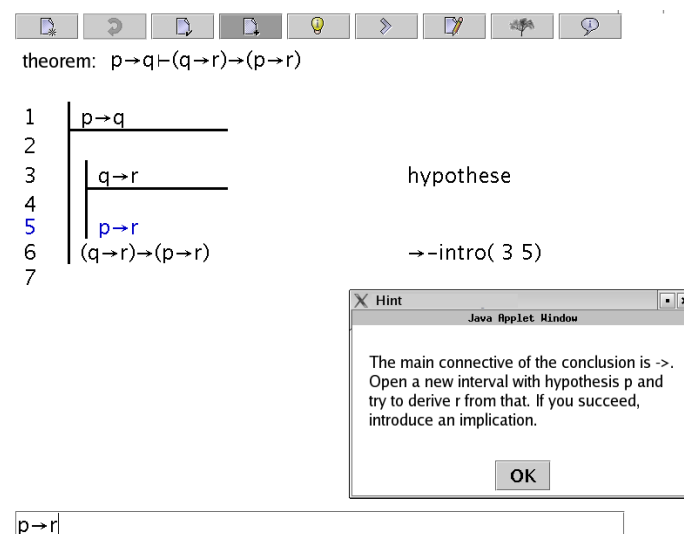


Figure 1: A screenshot of the BOP program. A student is trying to prove a theorem in propositional logic and has just pressed the HINT button. A popup window displays a hint which will help the student complete his prove of this theorem.

- When the student is unable to construct the proof, i.e. when he or she gets stuck at some point in the proof, he or she can use three functions to continue making progress:
  - HINT: ask the theorem prover to provide a textual hint.
  - STEP: ask the theorem prover to provide the next step in a correct proof.
  - PROOF: ask the theorem prover to provide the complete proof.

The goal of Bop is to train students in completing a proof. The theorem prover communicates with the client to make sure that the hints and steps provided lead to a correct proof. The hints provided to the student are based on the rules which the prover uses to infer a proof. The underlying algorithm is inspired by [2]. We show that the technique of searching in a space of instructive hints is a simple, yet effective method for tutoring logic. In the second-year course it turned out that students used the program mainly to practice for an exam; they found the program very useful and easy to use. Figure 1 shows a screenshot of the BOP program.

### 3 The Event

We would like to demonstrate this program to teachers and assistants of courses in logic. People who would like to participate should have some knowledge of propositional logic.

When you come to this event you will not only get to know Bop, and learn to construct proofs in the system of Fitch yourself, but you will also get some insight in how to use an automatic theorem prover for teaching how to proof. We will show the idea behind the web-interface, and the technical details of checking proofs and providing hints.

The event consists of:

- an introduction to the tool, including an example (10 minutes),
- 20 minutes in which each participant can try to prove a couple of theorems using this tutoring tool (at a computer), and in which they can fill in a questionnaire,
- an explanation of the technical design of the tool (10 minutes), and
- a discussion based on a questionnaire (5 minutes).

About 20 to 30 participants can join this event per session.

## References

- [1] Fredric Brenton Fitch. *Symbolic Logic*. the Ronald Press Company, New York, 1952.
- [2] Frederic D. Portoraro and Robert E. Tully. *Logic with SYMLOG - Learning Symbolic Logic by Computer*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1994.
- [3] Sicco E. Verwer and Jonne Zutt. Bop, a tutoring system for theorem proving in Fitch, 2004. <http://logica.ewi.tudelft.nl/>.



# Elements of ITS development technology and evaluation of its effectiveness

Ildar GALEEV, Oleg KOLOSOV, Alexander FILYAEV  
IER Department, Kazan State Technological University,  
K. Marx St., 68, Kazan 420015. Tatarstan, Russia  
E-mail [monap@kstu.ru](mailto:monap@kstu.ru)

**Abstract.** Effectiveness of intelligent tutoring systems (ITS) and the scale of their multiplication are defined by quality of their learning models that provide adaptive control of learning. Many ITS provide adaptive mastering of knowledge by students and work on the base of overlay model of student's knowledge, represented by pairs «concept-meaning». MONAP tools are meant for development of ITS that provide adaptive mastering of skills by students and work on the base of overlay model of student's skills, represented by pairs «rule (operation) - meaning». A learning process is considered to be a controlled process of solution of learning tasks.

## 1. General description of Interactive Event

The proposed Interactive Event (IE) is aimed at research of elements of ITS development technology with MONAP tools and evaluation of its effectiveness.

Evaluation of effectiveness of this technology will be made by IE participants according to the following criteria:

- universality of ITS development technology;
- laboriousness of ITS development;
- the level of skill needed for tutor – ITS developer;
- adequacy of learning model made by this technology.

The first three criteria need no comments. Of course, effectiveness of technology is higher as the limit for education domains and laboriousness of ITS development are lower; a small level for skill needed for ITS developer is also needed.

The main requirement for the models is their adequacy. A model is adequate when it reflects an object's behaviour with acceptable accuracy. Accuracy is defined as the level of similarity of the values of output parameter of model and object. A control action (in particular, a learning task) composed by the tutor for the next learning step in accordance with the level of student's skill is the output parameter for the learning model. Because of absence of "ideal" theory of education and quantitative characteristics (of evaluation) of "ideal" process of learning, it is impossible, for example, to make quantitative assessment of needed level of values of coincidence of output parameters of model and object (tutor) in general. Adequacy analysis of the model is qualitative and it is one of the main results of IE.

The IE target group is the experts in development and in use of ITS.

No special prior knowledge and skills are necessary. The target group defines needs for level of proficiency.

Two basic components can be defined in the object and in the model:

- evaluation of student's skills;

- forming of control action on the base of skills assessment.

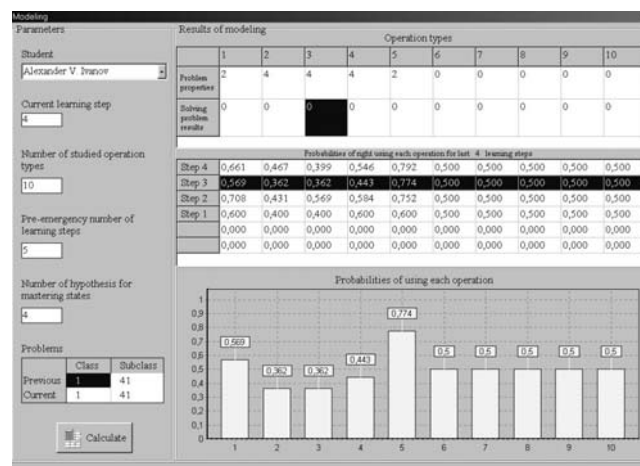
When designing a particular ITS, a teacher has to set values of the learning control subsystem parameters (Fig. 1).

**Figure 1.** Parameters of learning control subsystem

Approachment of skills assessment made by this model and evaluation made by tutor can be provided by the choice of corresponding dimension of estimate scale. Dimension of estimate scale is defined by the number of introduced mastering levels (Fig.1).

Because of the fact that the aim (output parameter) of the learning model and object (tutor) is not skills evaluation, but learning control on its base, the central problem is not to minimize unbalance in assessment, but to minimize differences in learning process control caused by this unbalance. From the tutor's point of view, unbalance in assessment can occur when evaluation of student's knowledge is either raised too high or set too low. When the tutor considers the estimate to be too high the value of optimal measure of difficulty should be lessened (Fig. 1) in order to lessen the difficulty of the learning task made by the model. Otherwise, when the tutor considers the estimate to be too low the value of optimal measure of difficulty should be raised. This method of minimization of the unbalance results in knowledge assessment between learning model and tutor is one of ways to provide adequacy of model for the tutor. Also, this adequacy can be achieved by flexible variation of the values of all other parameters of the model (Fig. 1).

To carry out IE, the modelling subsystem – a part of MONAP tools - will be used. In this mode (Fig. 2), a teacher can simulate solutions of the problems.



**Figure 2.** Modelling mode (diagram 1)

The teacher can see the results of knowledge identification process (Bayesian-based approach) in digital and graphic forms. In addition he/she is informed about a type of the next problem, which will be proposed for the student according to his/her mastering level and values of learning model parameters.

By modifying the results of problem solution and values of the learning model parameters the teacher is able to assort the appropriate values of the parameters for the optimal learning process control. In the modelling mode, two different forms for graphical description of the results of student's knowledge presentation are available. In the digital form, this information is placed in the table above the diagram. The first type of the diagram (fig. 2) presents probabilities of right use of each operation by the student on the defined number steps, i.e. presenting the data from chosen row of the table in a graphical way. The second type of the diagram (fig. 3) presents the history of mastering level changing for a particular operation (the graphical presentation of the chosen column of the table). To switch between these presentation modes the teacher has to click on the caption of the appropriate row or column correspondingly.

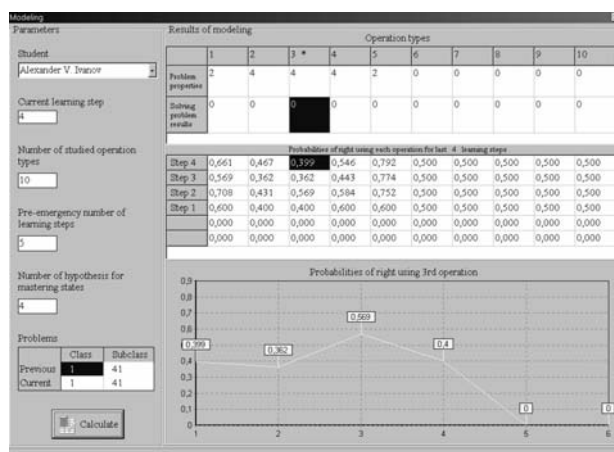


Figure 3. Modelling subsystem (diagram 2)

In connection with multifactorial, dynamic and feebly formalised character of learning process there is a need to have tools that provide the teacher possibilities to make sequential didactic experimentation. The aim of such experiments is to tune learning control model, which provides to allow particular conditions of designed ITS using. Modelling subsystem realised in MONAP meets the indicated requirements.

## 2. The working scenario

The working scenario of the IE can be described by the following steps:

- Installation of the modeling subsystem on PCs of participants of the IE (by IE organizers);
- Presentation of ITS development technology with MONAP tools (by IE organizers);
- Questions and answers on the technology and use of the modeling subsystem (by IE organizers and participants);
- Modeling of a learning process (by IE participants) for three student categories:
  - a) “excellent students” – do all exercises without mistakes, i. e.  $\forall j L_j = M_j$ ; (1)
  - b) “underachieving student” – all operations needed to do the exercise are done with mistakes, i. e.  $\forall j M_j = 0$ ; (2)

c) “normal student” – do some operations with mistakes, i. e.  $\exists j L_j \neq M_j$ . (3)

In the above formulas (1-3) the following symbols are used:

$L_j$  is the number of  $y_j$  operations necessary for solving the task ( $j=1,2,\dots,J$ );

$M_j$  is the number of correct  $j$ -operations while doing the exercise.

During the IE, the modeling subsystem described above (fig. 2, fig. 3) will be used - as exe-file. Also the technology for ITS development with MONAP tools will be presented. It is available at <http://ittal.kstu.ru>.

After the modeling process, the following topics are planned to be discussed:

1. What is the effectiveness of the technology for ITS development (labouriousness, universality, friendliness, and adequacy)?
2. Are there any more effective technologies for ITS development for adaptive control of skills acquisition?
3. What are the future cooperation prospects, aimed at:
  - development of integrated tools that will include MONAP ([http://ittal.kstu.ru/pr\\_monap.html](http://ittal.kstu.ru/pr_monap.html)) and partners' tools;
  - choice of topical domains for education and development of adaptive intelligent learning spaces that will include e-textbooks and ITS.

The authors' paper “Elements of ITS development technology and evaluation of its effectiveness” was submitted to the main conference.

### 3. Detailed description of the Technical Requirements to run the Interactive Event

Duration = (20+15+30+15+10) = 1 hour 30 minutes

Presentation of ITS development technology = 20 minutes

Questions on the technology = 15 minutes

Modeling = 30 minutes

Discussion of modeling results = 15 minutes

Discussion of possibility of cooperative projects realization = 10 minutes

Max number of participants is limited only by the number of PCs in the room.

Working settings: group, but also can be individual.

Special equipment is not needed (only PC and projection device for presentation).

The modeling subsystem should be installed on each PC.

### References

- [1] Brusilovsky, P. Methods And Techniques Of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 6 (2-3), 1996, pp. 87-129
- [2] I. Galeev. Automation of the ETS Desing, Educational Technology - September-October 1999. - V. XXXIX, No. 5. - pp. 11-15.
- [3] Galeev I., Chepegin V. and Sosnovsky S., MONAP: Models, Methods and Applications, Proceedings of the International Conference KBCS 2000, Mumbai, India. pp. 217-228.
- [5] Ildar Galeev, Larissa Tararina, Oleg Kolosov, Vlad Kolosov, Structure and implementation of partially integrated adaptive learning environment, in Allison Rossett (ed): Proceedings of E-Learn 2003, Phoenix, Arizona USA, November 7-11, 2003, p. 2151-2154.
- [6] Ildar Galeev, Larissa Tararina and Oleg Kolosov, “Development of software tools for intelligent tutoring systems”, in Lora Aroyo and Carlo Tasso (ed): Workshop Proceedings (Part I) of 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Eindhoven, The Netherlands, 23rd August 2004, p. 14-19.