Andes: A Coached Problem Solving Environment for Physics *

Abigail S. Gertner[1] and Kurt VanLehn[2]

[1] The MITRE Corporation, Bedford, MA 01730
[2] LRDC, University of Pittsburgh, Pittsburgh, PA 15260

Abstract. Andes is an Intelligent Tutoring System for introductory college physics. The fundamental principles underlying the design of Andes are: (1) encourage the student to construct new knowledge by providing hints that require them to derive most of the solution on their own, (2) facilitate transfer from the system by making the interface as much like a piece of paper as possible, (3) give immediate feedback after each action to maximize the opportunities for learning and minimize the amount of time spent going down wrong paths, and (4) give the student flexibility in the order in which actions are performed, and allow them to skip steps when appropriate. This paper gives an overview of Andes, focusing on the overall architecture and the student's experience using the system.

1 Introduction

This paper is an overview of problem solving in Andes – an Intelligent Tutoring System for introductory college physics. Andes interacts with students using *coached problem solving* [12], a method of teaching cognitive skills in which the tutor and the student collaborate to solve problems. In coached problem solving, the initiative in the student-tutor interaction changes according to the progress being made. As long as the student proceeds along a correct solution, the tutor merely indicates agreement with each step. When the student gets stuck or makes an error, the tutor helps the student overcome the impasse by providing hints that lead the student back to the correct solution path.

The fundamental principles underlying the design of Andes are: (1) encourage the student to construct new knowledge by providing hints that require them to derive most of the solution on their own, (2) facilitate transfer from the system by making the interface as much like a piece of paper as possible, (3) give immediate feedback after each action to maximize the opportunities for learning and minimize the amount of time spent going down wrong paths, and (4) give the student flexibility in the order in which actions are performed, and allow them to

---

* This research was supported by ARPA's Computer Aided Education and Training Initiative under grant N660001-95-C-8367, by ONR's Cognitive Science Division under grant N00014-96-1-0260, and by AFOSR's Artificial Intelligence Division under grant F49620-96-1-0180. The authors would like to thank the members of the Andes group.

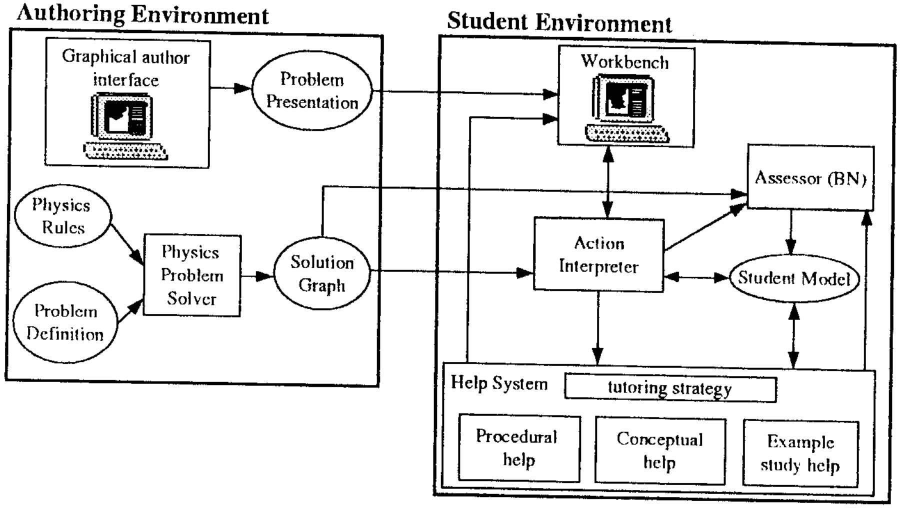


Fig. 1. The Andes System Architecture

skip steps when appropriate. This paper gives an overview of the system that we designed following these principles, focusing on the overall architecture and the student's experience using Andes. Several of the modules that Andes comprises are described in other papers and so we will not discuss them in detail here. In particular, we will not talk at all about the Self-Explanation coach or the Conceptual help system, which are described in [4, 5] and [1] respectively.

The following section provides a brief summary of the Andes architecture and implementation. Section 3 gives an example of the typical student interaction with Andes while solving a problem. In Section 4, we describe the underlying system for providing feedback and help. Finally, in Section 5 we present some of the work we have done to evaluate Andes with students.

2 Andes system overview

The Andes project began in September 1995, and is a collaboration between the University of Pittsburgh and the US Naval Academy. Andes is implemented in Allegro Common Lisp and Microsoft Visual C++ and runs on Pentium PCs under Windows 95.

Andes has a modular architecture, as shown in Figure 1. The left side of Figure 1 shows the authoring environment for creating new problems. Prior to run time, a problem author creates both the graphical description of the problem, and the corresponding coded problem definition. Andes' problem solver uses this definition to automatically generate a model of the problem solution space called the *solution graph.*

The right side of the figure shows the run-time student environment. The Workbench is the graphical interface with which the student studies examples and solves physics problems. The Workbench communicates with the Action Interpreter, which looks up the student's entries in the solution graph and provides immediate feedback as to whether the entries are correct or incorrect. More detailed feedback is provided by Andes' Help System [12]. Both the Action Interpreter and the Help System refer to the student model to make decisions about what kind of feedback and help to give the student. The central component of the student model is a Bayesian network that is constructed and updated by the Assessor, and provides probabilistic estimates of the student's mental state [3]. The student model also contains information about what problems the student has worked on, what interface features they have used, and what help they have received from the system in the past.

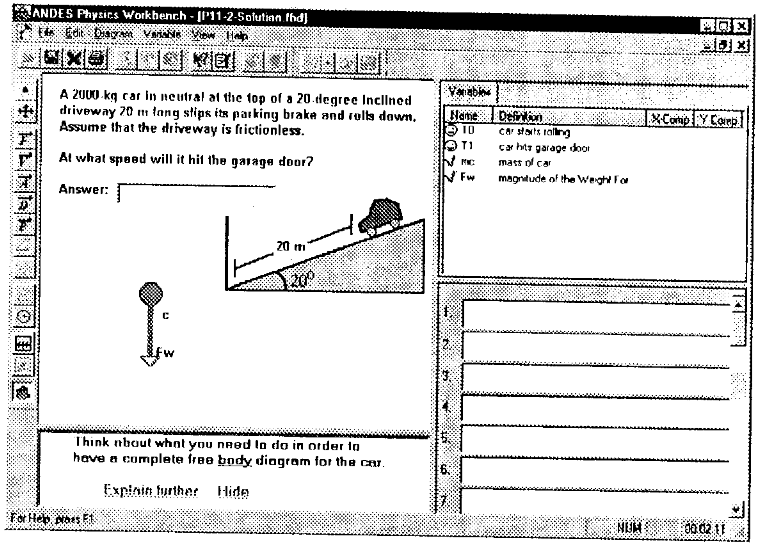
3 Solving problems with Andes: an example

One of the principles underlying the design of Andes was that the interface ought to be as much like a piece of paper as possible, so as to facilitate fading of tutorial support as students become more familiar with the physics domain, and the eventual transfer from solving problems with Andes to solving problems on paper. We attempted to keep the number of structured entry fields to a minimum, since every piece of structure in the interface might serve as scaffolding to the student, on which they could become dependent. As a result, the interface initially appears quite simple, consisting of two main entry panes (Figure 2), in which students can draw diagrams (upper left) and enter equations (lower right), as well as the variable definitions pane, located above the equation pane, and the hint window, below the diagram pane on the left.

3.1 Drawing diagrams

When a problem is first opened, the diagram pane contains a statement of the problem and a (read-only) picture of the problem situation. The lower part of the pane is initially blank. This area is provided for the student to perform a *qualitative analysis* of the problem before they begin working out the quantitative solution (in fact, some problems only ask for the qualitative analysis and do not require the student to write any equations). This type of qualitative reasoning is an important part of physics problem solving. It is used by expert physicists both in talking about simple physics problems [2] and when discussing real-world research results [10]. In addition, requiring qualitative reasoning in solving problems has been found to uncover students' misconceptions better than allowing students to use algebraic reasoning alone [11, 8].

Students use the drawing tools to the left of the diagram pane to enter elements of a free body diagram such as force vectors. The student can also draw motion vectors such as velocity and acceleration. To draw a vector, the student clicks and drags the mouse in the diagram pane, rotating the vector until it points in the desired direction and then releasing the mouse. Other items that can be included in a diagram include coordinate axis systems, angles between



A 2000 kg car in neutral at the top of a 20 degree inclined driveway 20 m long slips its parking brake and rolls down. Assume that the driveway is frictionless.

At what speed will it hit the garage door?

Answer:

Name	Definition	X-Comp	Y Comp
T0	car starts rolling		
T1	car hits garage door		
mc	mass of car		
Fw	magnitude of the Weight For		

Think about what you need to do in order to have a complete free body diagram for the car.

Explain further Hide

Fig. 2. The Andes problem solving interface

vectors and axes, and the radius of a circular path. All of these are drawn with the mouse using the tools on the left of the screen.

After the student has drawn an item in the diagram window using the mouse, a dialog box appears in which the student must enter information defining the object they have just drawn. This departure from the "piece of paper" principle is necessary so that Andes can give appropriate feedback based on not just what the student drew (eg. a vector pointing straight up) but also what they *meant* by that entry (eg. a normal force exerted by the driveway on the car).

3.2 Variable definitions

Another significant way that the Andes interface differs from a piece of paper is in the definition of variables and how they are used in equations. When a student is solving a physics problem on paper, she can start out right away by writing down an equation, such as $\mathbf{F} = m * \mathbf{a}$ without explicitly stating what $\mathbf{F}$, m, and $\mathbf{a}$ refer to. Andes, on the other hand, requires all quantities to be defined explicitly before they may be entered in equations. This enforces a systematic approach to problem solving which can greatly reduce the number of careless mistakes students make.

Variables may either be defined by assigning a label to an item in the diagram, or by using a special purpose variable definition menu. Certain quantities (such as scalar quantities like speed) cannot be drawn in a diagram, so the variable menu must be used to define variables for these quantities. Using the variable menu involves choosing the type of quantity to be defined and then filling in a dialog box similar to the ones for diagram entries.

Whether a variable was defined as a diagram entry or using the variable menu, after it has been defined it will appear in the variable definitions pane. This pane lists all defined variables with their definitions and, for vectors, the names of their components.

3.3 Entering equations

Equations are entered in the text fields in the lower right pane of the Workbench, a single equation per line. There is no structured equation editor in Andes. Students use a conventional syntax for entering equations (operators *, /, +, and -; ^ for exponentiation; _ for subscripts). Students can enter anything they want into the equation field and Andes will attempt to parse it and determine whether it corresponds to a correct equation [6]. There may be more than one possible parse for an equation, in which case Andes will look up whether any of the parses represents a correct equation.

Variables in equations must first be defined so that they are listed in the variable pane. If Andes finds tokens in the equation that can only be interpreted as undefined variables, it displays an error message listing those tokens. If the equation cannot be parsed for some other reason, it displays an error message informing the student that it could not interpret the entry. Otherwise, Andes gives simple correct/incorrect feedback on the equation entry.

4 Feedback and help

As illustrated in Figure 1, the Action Interpreter is the central module of Andes. It is responsible for getting student input from the Workbench, recording the input in its databases, and returning feedback to the Workbench which includes information about whether the student's action was correct or incorrect, as well as hints and help messages to be displayed. The following sections describe how the Action interpreter and related modules do their jobs.

4.1 The student model

Modeling a student in an ITS involves a great deal of inherent uncertainty regarding not only the student's *beliefs* and *goals*, but also the level of *knowledge* that she has about the domain. There is additional uncertainty in Andes since the student is not constrained to perform actions in a particular order. This means that even if Andes has not observed the student performing a certain action, it cannot assume that the student doesn't know *how* to perform that action because the student may intend to perform it in the future. To address these multiple sources of uncertainty, Andes' student model combines information about the current state of the problem solving process with long-term assessment of the student's knowledge of physics in a probabilistic representation using Bayesian networks [3].

Each physics problem is represented by a separate Bayesian network. As a student moves from one problem to the next, Andes' assessment of her general physics knowledge is updated and used to initialize the model for the next problem. Thus the modeling of ***problem-specific*** knowledge about each problem is related through the ***domain-general*** assessment of physics knowledge that the problems have in common.

The Bayesian network for each problem is constructed from a data structure called a *solution graph* which represents alternative solution paths for each problem, including some that involve "buggy" rules and thus represent typical incorrect student solutions. Andes' Bayesian networks may have anywhere from around 100 to over 200 nodes, depending on the complexity of the problem. Using a network that represents the entire solution space for the problem means that the student model always has estimated probability values, even for steps that haven't been explicitly observed. This supports the Andes design principle of allowing the student maximum flexibility in performing solution steps in any order and skipping steps.

Bayesian networks provide us with a principled framework for combining all of the different sources of uncertainty about the student's problem-solving processes. For example, if a given fact may be derived in two different ways, and a student is observed to know the fact, the Bayesian network provides a straightforward way of sharing the credit for that knowledge between the two alternative derivations. Furthermore, if one of the derivations is known to be less likely than the other (e.g., because it depends on a rule that the student probably does not know), then the credit will be apportioned accordingly – giving more weight to the derivation that the student is likely to know.

The Action Interpreter interacts with the Bayesian network in two ways. First, when Andes observes the student performing an action that corresponds to a node in the Bayesian network, the value of that node is clamped to True, and the network is updated to reflect the new evidence. Second, when Andes needs to respond to the student in a way that depends on some aspect of her current knowledge or mental state, the Action Interpreter queries the Bayesian network to find out the current probabilities associated with the relevant nodes. This can be used for resolving ambiguities in determining the student's current goal, as well as for determining the appropriate level of help to give for part of the problem.

4.2 Immediate feedback

When a student makes an entry in Andes, the workbench always responds with immediate flag feedback by changing the color of the entry – green for correct and red for incorrect. This feedback is generated using information from the solution graph. Here, we will describe how the feedback is generated for diagram entries and variable definitions. Feedback for equation entries is more complicated, and is described in some detail in [6].

Each student entry may reflect several different pieces of information, as represented by the multiple fields of the dialog boxes for defining diagram entries. For example, the definition of a force vector includes fields for the type of force, the object it is acting on, the agent of the force, and its direction. When the student enters a definition, the Action interpreter looks at the information in the solution graph to see if any quantity exists there with exactly the same features as the student's entry. If it finds such an entry, the dialog box disappears and the entry turns green on the screen. If no matching quantity is found, the Action Interpreter has to determine what part of the entry is incorrect. To do this, it attempts to determine what the student was most likely to have been trying to enter, and then finds the features that differ between that intended entry and the actual entry.

Finding the most likely intended entry is done using a combination of matching features and probabilities. The solution graph quantity with the most features in common with the student's entry, and the highest probability in the Bayesian network, is selected as the intended entry. This entry is then compared to the student's entry to finds those features that differ between the two. Andes then turns red only the fields in the dialog box corresponding to the mismatched features. This gives the student specific feedback on what part of their entry was in error.

4.3 What's wrong with that?

As noted earlier, one of the design principles for Andes was to encourage *constructive*, as opposed to passive, learning. Therefore, Andes gives away as little information as possible unless the student asks for it. In the case of errors, Andes always starts by giving flag feedback. This feedback is accompanied by a hint or error message only in the case of simple syntactic errors. For all other errors, the student is expected to attempt to fix the problem on her own if she can.

If the student is not able to fix an erroneous entry on her own, she can select the entry and ask "what's wrong with that?" using a menu, and Andes will respond with a short hint intended to point the student toward the feature of her entry that was incorrect. For example, if the entry was an acceleration vector labeled 'a' whose direction was incorrect, the hint might say "think about the direction of 'a'."

When a non-specific hint is given, there will also be a link labelled "Explain Further" in the hint window, which the student can click on to get more information. Clicking repeatedly on "Explain Further" will eventually result in a hint that explicitly tells the student how to fix their entry. For example, the bottom-out hint for the direction of the acceleration vector might be "The direction of 'a' is horizontal and to the left."

4.4 Procedural help

Procedural help is the part of Andes' help system that is responsible for generating a hint when the student gets stuck and asks for help [7]. Hints are generated based on the state of the student model at the time the student asks for help. To produce a hint, Andes first selects a node in the solution graph that will be the topic of the hint. The topic node should represent a proposition that is relevant to what the student has been doing recently and that the student is likely to want to address next. The hint topic node is selected by first identifying a goal node in the solution graph that is likely to be the reason for the student's most recent action, and then following a path from that goal to find a node representing an action that the student has not yet done and, according to the Assessor, probably does not know about (this procedure is described in more detail in [7]).

Once a hint topic node has been selected, the hint template belonging to the proposition the node represents is instantiated with the contextually relevant information from the problem to generate an English string to be displayed to the student. As in the case of the "what's wrong with that" hints, the initial hint given for a particular topic node will be quite general, to encourage students to recover from the impasse on their own. Students can then click on a link to get successively more specific hints until they are able to continue solving the problem.

5 Evaluations of Andes

Andes' development has been carried out in conjunction with an ongoing series of formative evaluations, which were used to assess the usability of the interface, suggest new features, and evaluate the effectiveness and clarity of hints and help messages. Prototype versions of the system have been used by students at the US Naval Academy every semester since the Fall of 1996. Additionally, several more in-depth user studies were carried out at the University of Pittsburgh during that time.

At the Naval Academy, students taking the introductory physics course were asked to use Andes to do some of their homework. To do this, each student had to download the Andes installation file from the local network, install it on their personal computer in their dorm room, and start using on their own after just a short demonstration. Right away, this required us to implement an easily used installation program, to make sure that the Andes interface was simple enough that students could get up and running without much help, and to include extensive on-line help files that they could access while they were using the program.

We used several methods for getting information from students during these formative evaluations. First, they were encouraged to enter comments in a special dialog box as they worked with Andes. We found that students did not enter comments very often, but when they did we were able to bring them to the attention of the instructors who would quickly address the student's question or concern. Second, students filled out a questionnaire about their experience using the program. These questionnaires turned out to be not very informative because students did not give much detailed information about what they would like to see changed in the system. Third, they communicated frequently with their instructors about problems they were having using the system. This was extremely important as it allowed the instructors to develop a very complete overall view of the problems people were having and what we could do to fix them. Finally, all student activity within Andes is recorded in log files which the students were prompted to upload each time they exited the program. These log files, including students' comments as well as a complete record of every action performed on the interface, provided a wealth of information to guide us in improving the system. Log files were also used by the instructors to get a better sense of what a student had been doing when they came in for extra help.

At the University of Pittsburgh our approach in evaluating the system was to record a small number of sessions with students using Andes, and to have the students "think out loud" as they worked with the program. We could then identify significant events in the session records where students failed to learn something due to a flaw in Andes' help, and rank them according to frequency and importance. We used these events as targets for improving the system and tested our changes by performing the same actions on the new version of Andes and seeing if the help it gave had improved. In this way, we were able to fix many of the major problems students had both with the interface design and the help system.

In the Fall of 1999 we performed a summative field evaluation of Andes at the Naval Academy. Andes was used for four weeks by 173 students in eight sections of the Naval Academy's introductory physics course. At the end of this time, they were given a midterm exam covering material that was taught by Andes (and by instructors during course lectures and sections). The students' performance on the midterm was compared to a control group of 162 students whose sections did not use Andes. The results of this comparison were encouraging. Students who used Andes performed 2.9 percent (1/3 of a letter grade) better on average than students who did not use Andes ($p \leq 0.033$). This compares favorably with other successfully evaluated tutoring systems. For instance, the PUMP Algebra Tutor had an effect size[1] of 0.3σ on standardized math tests [9], whereas Andes effect size was 0.2σ on the normal exam used by the whole course. Although the PUMP Algebra Tutor also had an effect size of approximately 1.0σ on tests designed for its content, we did not use such tests in our evaluation of Andes.

Perhaps more interestingly, when the Andes results are broken down by the students' major, we see that the effect of Andes for humanities majors was the largest (7.3 percent), with the effect for science majors next largest (3.9 percent). In the case of the engineering students (who were also taking a course on statics concurrently) there was actually a small (1.3 percent) negative effect for Andes. Thus, Andes appears to be most effective with students who are most likely to need help learning physics.

6 Conclusions

In this paper we have provided an overview of the Andes system, showing how we were guided by the four design principles listed in the Introduction. We have

---

[1] Effect size is calculated as the difference between the score of the experimental group and the score of the control group, divided by the standard deviation of the control.

learned a great deal in the process of developing and deploying Andes, both about designing an ITS to teach complex problem-solving skills and about the integration of such a system into an existing instructional environment. Future work on this project should continue to yield many more insights (see [13] in this volume).

References

[1] P. L. Albacete and K. A. VanLehn. The conceptual helper: An intelligent tutoring system for teaching fundamental physics concepts. In *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems*, 2000.

[2] M. Chi, P. Feltovich, and R. Glaser. Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5:121–152, 1981.

[3] C. Conati, A. S. Gertner, K. VanLehn, and M. J. Druzdzel. On-line student modeling for coached problem solving using Bayesian networks. In *Proceedings of UM-97, Sixth International Conference on User Modeling*, pages 231–242, Sardinia, Italy, June 1997. Springer.

[4] C. Conati and K. VanLehn. Teaching meta-cognitive skills: implementation and evaluation of a tutoring system to guide self-explanation while learning from examples. In *In ¿ Proceedings of AIED 99, 9th World Conference of Artificial Intelligence and Education, Le Man, France*, 1999.

[5] C. Conati and K. A. VanLehn. Further results from the evaluation of an intelligent computer tutor to coach self-explanation. In *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems*, 2000.

[6] A. S. Gertner. Providing feedback to equation entries in an intelligent tutoring system for physics. In *Proceedings of the 4th International Conference on Intelligent Tutoring Systems*, San Antonio, August 1998.

[7] A. S. Gertner, C. Conati, and K. VanLehn. Procedural help in Andes: Generating hints using a Bayesian network student model. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.

[8] J. I. Heller and F. Reif. Prescribing effective human problem-solving processes: Problem descriptions in physics. *Cognition and Instruction*, 1(2):177–216, 1984.

[9] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. In J. Greer, editor, *Proceedings of the 7th World Conference on Artificial Intelligence and Education*, pages 421–428, Charlottesville, NC, 1995.

[10] A. Van Heuvelen. Learning to think like a physicist: A review of research-based instructional strategies. *American Journal of Physics*, 59(10):891–897, 1991.

[11] A. Van Heuvelen. Overview, case study physics. *American Journal of Physics*, 59(10):898–907, 1991.

[12] K. VanLehn. Conceptual and meta learning during coached problem solving. In C. Frasson, G. Gauthier, and A. Lesgold, editors, *Proceedings of the Third International Conference on Intelligent Tutoring Systems ITS '96*, pages 29–47. Springer, 1996.

[13] K. VanLehn, R. Freedman, P. Jordan, R. C. Murray, R. Osan, M. Ringenberg, C. Rosé, K. Schulze, R. Shelby, D. Treacy, A. Weinstein, and M. Wintersgill. Fading and deepening: The next steps for andes and other model-tracing tutors. In *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems*, 2000.