

Reflections on Andes' Goal-free User Interface

Kurt VanLehn¹

Published online: 16 October 2015

© International Artificial Intelligence in Education Society 2015

Abstract Although the Andes project produced many results over its 18 years of activity, this commentary focuses on its contributions to understanding how a goal-free user interface impacts the overall design and performance of a step-based tutoring system. Whereas a goal-aligned user interface displays relevant goals as blank boxes or empty locations that the student needs to fill with specific content, a goal-free user interface is essentially a blank canvas, with no visual indications of the goals that should be attempted next. This commentary also briefly mentions work that occurred after the “final” report on Andes appeared in this journal in 2005. The newer work focused on getting students to ask for hints when they need them.

Keywords Intelligent tutoring system · Physics education research · Intelligent user interfaces

It is a privilege to be asked to comment retrospectively on the Andes physics tutoring system project (VanLehn et al. 2005). When the Andes project began in 1996, it was strongly influenced by the CMU tutoring systems, which were already quite mature (Anderson et al. 1995). The original vision for Andes kept some of the main design elements of the CMU tutoring systems: immediate feedback, hint sequences, model-tracing and embedded assessment (student modeling) at the granularity of individual rules. Like the CMU tutors, it supported a whole year of instruction and was aligned with common physics curricula. However, all the CMU tutors employed user interfaces that could be called *goal-aligned forms*. At any point in time, the set of possible user actions were strongly constrained, such as picking from a fairly short menu or entering fairly constrained text into a blank. These actions were aligned with problem solving goals, in accord with Principle 2 of Anderson et al. (1995) which advocated communicating the goal structure of the problem via the user interface. Although goal-aligned forms seemed like appropriate scaffolding for novices, they also seemed too inflexible

✉ Kurt VanLehn
Kurt.VanLehn@asu.edu

¹ Arizona State University, Tempe, AZ, USA

because the scaffolding could not be easily faded as the student's competence grew. According to Singley and Anderson (1989), fading (i.e., gradually removing) the goal scaffolding should increase transfer to solving problems without the tutor.

In contrast, Andes was intended to be like paper in that the user could write anything anywhere on the page. Unlike paper, Andes would recognize both the user's intended goal and the user's success at achieving that intent. Thus, what set Andes apart from other tutoring systems of the mid-1990s was its commitment to a goal-free user interface.

The switch from goal-aligned forms to a goal-free user interface seemed simple technically, but it turned out to have surprising repercussions. As tutoring system designers today can even more easily create goal-free interfaces on tablets and touchscreens, it is worth reviewing what was learned from Andes' experiment with a goal-free interface. However, before discussing the lessons learned, it is worth a brief review of the relevant features of Andes.

The Goal-Free User Interface of Andes

Andes tutored physics problem solving as done in introductory college and advanced high school courses. When solving a physics problem on paper, the student does two main types of steps: writing an equation and defining a variable. When a vector variable is defined, the user may or may not draw the vector. Occasionally students make other inscriptions, such as drawing an ideal body, drawing coordinate axes or drawing arrows for defining the current direction in a circuit. However, the majority of the steps in physics problem solving are writing equations and defining variables. When working on paper, students are encouraged to enter their steps in a logical sequence, but a solution that has steps strewn about the page or missing is not really wrong if all the steps are true statements about the situation. Andes was intended to allow the same freedom, so users could enter steps anywhere and in any order.

The initial user interface for Andes only partially implemented a goal-free interface. Students could enter any equation in any location, but when they defined a variable, they had to use a goal-aligned form (see Fig. 1 of VanLehn et al. 2005). It wasn't until Andes3 was released around 2010 that the goal-free interface was finally finished and all forms were banished. The final user interface (see Fig. 1) was designed to look and act like MS PowerPoint, with tools for entering text, equations, arrows, coordinate axes, and other glyphs (Ranganathan et al. 2014). To define a variable, the user opened a text box at any location on the screen and typed text such as, "Let i_2 be the current through R_2 ." Andes matched the text dynamically to a large set of expected definitions, offering auto-completions when it could. For instance, after the user typed "Let i_2 b" Andes would add letters to make it "Let i_2 be the". If the user then typed a "c" then Andes would add letter to make it "Let i_2 be the current through" and so on.

Lesson 1: Providing Goal Scaffolding in the Context of a Goal-Free User Interface

The first major issue raised by using a goal-free user interface was how to provide the goal scaffolding that goal-aligned forms provided. Instead of providing the scaffolding

A man pushes a 20.0 kg crate across a frictionless floor with a horizontal force of 24.0 N. What work is done by the man on the crate in displacing it by 5.00 m?

Answer:

Time T0: pushing begins.
Time T1: pushing ends.

Let d be the displacement of the crate

$d = 5$

Tutor 3000
Forgot to put units on a number.
[Explain more](#)

Fig. 1 Andes3 screenshot. The student's steps are in *green* and *red*. The student has just typed “ $d=5$ ” in a text box, pressed the Enter key, and received an unsolicited hint

permanently, Andes provided goal scaffolding only when the user asked for it or seemed to need it. In particular, Andes provided two hint buttons. The “next step” hint delivered the goal scaffolding as a sequence of hints. The “what’s wrong” hint also generated a sequence of hints, but the first hint tried to indicate what was wrong with the selected step. Subsequent hints produced the same goal scaffolding as the next-step hint.

At the time the Andes project began, an unquestioned assumption of step-based tutoring systems was that when the user asked for goal scaffolding, the system should recognize what goal the user was trying to achieve and provide hints about that goal. This meant that the system had to do *plan recognition*, which meant parsing the user’s past activity in terms of plans and goals in order to figure out which plans were incomplete and thus which goals were active for the user. Andes1 implemented plan recognition by generating a plan in advance using a rule-based expert model of how to solve physics problems. The plan was an acyclic directed graph of goals and rule applications. This was converted to a Bayesian network. Past actions taken by the user were interpreted by clamping the corresponding nodes in the Bayesian network and updating the network. A node’s posterior probability represented the chance that the user was trying to achieve the goal represented by that node when the user asked for help. Note that goal-aligned forms do not do such plan recognition, because the location where the user is trying to make an entry or menu selection indicates the goal the user needs help on.

The first big surprise was how poorly this Bayesian plan recognition worked for physics problem solving with real students. Performance was measured by taking 40 snapshots of students’ work at a point where they asked for help and asking physics instructors to indicate what goal they thought the students needed help on. Instructors disagree on 19 snapshots, but in 21 snapshots where they did agree, Andes agreed with the instructors on only 3 (VanLehn et al. 2005). The problem wasn’t in the Bayesian plan recognition technique, but in the assumption that students were following a plan. The instructors completely agreed that in most of the 40 cases, the students appeared to have no recognizable plan. Their advice in most cases was to ignore the students’ work and simply start walking the student through a correct plan. So that’s what Andes2 did. The Bayesian network that made Andes1 unique was abandoned.

Lesson 2: Interpreting Equations is Hard When Students Can Write Any Combination of Equations

The second big surprise was how difficult it was to interpret equations. Physics problem solving involves both mathematics and physics. A physics rule application generates a single equation, perhaps with some variable definitions as well. Mathematical rules combine equations and simplify them. In order to keep Andes' job tractable, it only provided advice and assessment on physics rules. However, students rarely entered just the primitive equations generated directly by physics rule applications. They almost always wrote equations that were combinations of several equations. Thus, the problem was to analyze a student equation such as $Fg = a * 5 \text{ kg}$ into its constituent equations, $F_{net_y} = m * a_y$, $-Fg_y = F_{net}$, $a_y = -a$, and $m = 5 \text{ kg}$. Andes1 tried to solve this problem by pre-generating all algebraic combinations of primitive equations. This exploded combinatorially.

Fortunately, a sabbatical visitor and physicist, Joel Shapiro, found an elegant mathematical algorithm that could decompose almost any composite equation into its primitive constituents (Shapiro 2005). The algorithm searched for a set of primitive equations such that a linear combination of their gradients at the solution point equaled the gradient of the student's equation at the solution point. This algorithm is one of the small treasures of ITS research and it should be more widely used.

Lesson 3: Goal-Free Interfaces Make Embedded Assessment Much Harder

The third big surprise was that the goal-free interface made it much more difficult to obtain accurate embedded assessments (also called student modeling). With its roots in the Olae assessment engine (Martin and VanLehn 1995; VanLehn and Martin 1998), the early phase of the Andes project was rather obsessed with embedded assessment (Conati et al. 2002; A. Gertner et al. 1998; A. S. Gertner and VanLehn 2000; Schulze et al. 2000a; Schulze et al. 2000b). Along with the CMU tutors, Andes assumed that mastery learning would cause large gains in either effectiveness or efficiency, so it was worth determining which rules a student had mastered. This would allow Andes to choose problems adaptively and to move the student on to the next unit only when the student was ready.

We mounted a large study using synthetic students in order to evaluate the Andes1 approach to embedded assessment (VanLehn and Zhendong 2001). This was probably one of the best papers to come out of the project, but it is seldom cited, so it is worth reviewing its main conclusions, which apply to doing embedded assessment of multi-step problem solving even without a Bayesian network.

1. Even though the system is an assessment system, it should give immediate feedback on steps so that students will stay on a recognizable solution path. If student errors are not corrected, it becomes difficult or impossible to recognize subsequent correct reasoning and assign the appropriate credit.
2. When a student makes a mistake, the system must infer what goals and rules the student was trying to apply. This allows it to assign the appropriate blame to the

rules that should have applied. Without such evidence against mastery, it is difficult to identify rules that need more practice.

3. The smaller the steps, in terms of the number of rule applications between adjacent steps, the higher the accuracy of the assessment given a fixed number of problems solved. For interpreting *incorrect* steps, step size always matters because the more rules that share the blame for the error, the more evidence it takes to narrow in on the weak rule(s). For interpreting *correct* steps, step size is only important when the task domain often affords multiple derivations of the correct steps; this was not the case with physics, so step size did not matter for interpretation of correct steps on Andes.

Andes1 provided immediate feedback (finding 1 above), but it could seldom infer the goals and rules behind an incorrect step (finding 2). In short, Andes1 was seldom able to assign blame to the rules that caused incorrect steps.

Although embedded assessments were an important research topic for the Andes group, our collaborators at the US Naval Academy were unable to use them. The assessments were intended to be used for selecting homework problems that would match a particular student's needs. However, the Naval Academy students would have objected strongly if different students were given different homework exercises. This would have been perceived as "unfair." Thus, mastery learning and other forms of adaptive problem selection could not be used. Moreover, because homework was done in unsupervised settings (e.g., at home), it could not be used for grading, advancement or placement in the competitive environment of the Naval Academy. Thus, the next version of Andes, Andes2, included no student modeling. Nonetheless, the learning gains reported in (VanLehn et al. 2005) were positive and impressive, especially given that they were done in real-world classrooms.

In order to support educational data mining, Adaeze Nwaigwe developed a heuristic solution to the assignment of blame problem. She studied several heuristics, and found that the best one agreed well ($\kappa=0.78$) with human coders (Nwaigwe et al. 2007). Andes3 adopted this heuristic (Van de Sande 2013), which is:

1. First apply specific, well-defined small edits (e.g., changing + to -) to the students' incorrect step. If an edit creates a correct step, then blame the rule associated with the edit.
2. If that fails, then see if the user subsequently edited the incorrect step and made it correct. If so, then blame the rules that generated the correct step. One of them was probably weak and caused the error.
3. When users do not correct an incorrect step, it is typically because they deleted it—they almost never leave an incorrect step (which is red in Andes) on the screen. However, when they subsequently enter a correct step that is the same type of user interface element (equation, vector, etc.), then blame the rules that generated the correct step.

Lesson 4: Students do not Ask for Hints Often Enough

In work completed after the 2005 report, we discovered that Andes' students did not use hints often enough. Muldner et al. (2011) found that when students received

negative feedback (their entry turned red), they were four times more likely to make another attempt at the entry than to ask for a hint. When verbal protocols of students using Andes were analyzed (Ranganathan et al. 2014), and an episode was defined as beginning when Andes turned the student's entry red (incorrect) and ending when the student either corrected the entry or deleted it, then it turned out that in 38 % of the episodes, the student never asked for help; in 52 % of the episodes, the students asked for help from Andes alone; and in the remaining 12 % of the episodes, the students asked for help from the experimenter and Andes. Although the episodes where the students asked for help often ended well, with the student either learning something new or remembering a relevant piece of knowledge they had forgotten, the episodes where students did not ask for help usually ended poorly, with students guessing repeatedly until they stumbled onto a correction without understanding it or gave up and deleted the entry.

This underuse of hints is not a property of Andes' goal-free user interface, because similar student behavior has been noted with goal-aligned forms and many other systems (Aleven et al. 2003). For instance, Aleven and Koedinger (2000) found that when students made a mistake entering an explanation into the Geometry Tutor, they were about ten times more likely to edit their explanation than to ask for a hint; for mistakes on numerical steps, they were twice as likely to try again than to ask for help. This underuse of hints suggests that giving learners total control over when they get hints may not always be the best policy, regardless of the user interface.

The most recent version of Andes (Ranganathan et al., unpublished) alleviated this problem by judiciously providing unsolicited hints (see Fig. 1) and unsolicited meta-hints. Unlike the meta-tutoring of the Help Tutor (Roll et al. 2011), Andes' meta-hints always addressed the same issue and used the same wording ("You should ask for a hint."). This raised the frequency of getting a step right on the first attempt by $d=0.682$ and raised the problem completion rate by $d=0.727$.

Summary

With these modifications, Andes finally achieved the vision of a goal-free user interface that provided almost all the functionality of a goal-aligned form. If anyone wants to build a goal-free user interface for their tutoring system (as the FACT project is doing; see fact.engineering.asu.edu), here are the lessons from Andes to keep in mind:

- Because a goal-free user interface doesn't impose a goal structure/plan on the users, they often do not have one when they ask for help. It is perhaps better to advise them to follow the next open step in an expert's plan rather than the next open step of a plan that some plan recognition algorithm thinks that the student is following, where a step is "open" if the student has not yet taken it.
- When a step is incorrect, and the system cannot recognize how the step was derived and yet it is important for assessment to blame some rule(s) for the incorrect step, then a goal-free user interface makes assignment of blame harder because the location of an incorrect step provides no information about the user's intended goal. Thus, it is perhaps best to encourage the user to edit the entry and make it correct, providing hints if asked. Analysis of the correct entry allows the system to

determine what the user was trying to do, which then allows the assessment system to blame the rules that initially prevented the user from achieving that goal.

- Unlike a goal-aligned form, a goal-free user interface does not control the size of steps, where the “size” of a step refers to the number of inferences undertaken between the preceding step and this one. Goal-free users often enter large steps, so the tutoring system must be equipped to decompose such large steps into their primitives.
- Regardless of whether the user interface is goal-free or a goal-aligned form, giving unsolicited meta-hints and hints may improve students’ use of the tutoring system.

Future Work

Although much has been learned about goal-free user interfaces for tutoring systems, there are still several unanswered questions.

Perhaps the main unanswered question is whether a goal-free user interface increases transfer from a goal-aligned user interface to paper. Given that novices often have no plans and that students often do not ask for help when they should, it seems prudent to require novices to use a goal-aligned form initially. However, as they internalize the plans and become more competent, would it be better to keep them practicing on a goal-aligned form or to give them a goal-free user interface like Andes’? Removing the goal-aligned form probably removes many of the contextual cues that are incorporated in the students’ knowledge, so memory theory suggests that goal-free user interfaces would increase transfer (Singley and Anderson 1989). However, this hypothesis remains untested.

Another unanswered question involves the use of natural language dialogues instead of goal-aligned forms for communicating the goal scaffolding. When Andes users asked for help and Andes decided to walk them through the expert’s plan, it used a scripted dialogue. This violates a principle of educational user interface design that one high school teacher and researcher expressed as: “Kids like to click; they don’t like to read.” (C. Chase, personal communication, 2013). When such students use a goal-free interface and ask for a hint, it may be better to provide a goal-aligned form than a dialogue. This would be consistent with the repeated findings of null effects in comparing hint dialogues to text that has exactly the same content (Dzikovska et al. 2014; Evens and Michael 2006; Siler et al. 2002; VanLehn et al. 2007; Weerasinghe and Mitrovic 2006). VanLehn (2011) discusses this finding further.

Yet another unanswered question is whether goal-free user interfaces might facilitate routine production of step-based tutoring systems. CTAT (Alevin et al. 2009) and similar authoring tools can be used to create goal-aligned user interfaces, but they are not easily used for creating a goal-free user interface. The Andes3 user interface is a lightweight web app that doesn’t know it is part of a physics tutoring system; it could be used with other task domains and systems. This suggests the technical possibility of having exactly the same user interface with a wide variety of tutoring systems—they all use the same client but the server code is different for different tutoring system. A general-purpose client would provide only syntactic support, such as parsing mathematical expressions and giving feedback if the typed input is ill-formed, or completing typed text by matching it against a set of expected strings.

The reader may wonder what happened to Andes. The server side of Andes3 is written in Lisp, and the client side is written in JavaScript using the Dojo library. The subset of the languages used by Andes code is so stable that the system has continued to run with almost no maintenance for several years. Until October, 2014, Brett van de Sande kept Andes3 running on a server at Arizona State University and responded to the occasional email from users. When he left to take another position, the server was turned off. Andes3 is open source, so it would in principle be easy to bring it up again. Please contact the author if you are interested.

Acknowledgments Any success that the Andes project had was due to the talents and efforts of Patricia Albacete, Winslow Burleson, Min Chi, Susan Chipman, Cristina Conati, Ellen Dugan, Abigail Gertner, Ken Koedinger, Robert Hausmann, Collin Lynch, Adaye Nwaigwe, Zhendong Niu, Kasia Muldner, Charles Murray, Rajagopalan Ranganathan, Michael Ringenber, Kay Schulze, Joel A. Shapiro, Robert Shelby, Stephanie Siler, Linwood Taylor, Don Treacy, Anders Weinstein, and Mary Wintersgill. Brett van de Sande deserves special recognition for leading the project from 2008 to 2014. The Andes project was supported by the Cognitive Science Program of the Office of Naval Research under grants N00019-03-1-0017 and ONR N00014-96-1-0260, and by NSF under grants 0354420 and 0836012.

References

- Aleven, V., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Intelligent tutoring systems: 5th International Conference, ITS 2000* (pp. 292–303). Berlin: Springer.
- Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R. M. (2003). Help seeking and help design in interactive learning environments. *Review of Educational Research, 73*(2), 277–320.
- Aleven, V., McLaren, B., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence and Education, 19*, 105–154.
- Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences, 4*(2), 167–207.
- Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User Adapted Interactions, 12*(4), 371–417.
- Dzikovska, M., Steinhäuser, N., Farrow, E., Moore, J. D., & Campbell, G. (2014). BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *International Journal of Artificial Intelligence and Education, 24*, 284–332.
- Evens, M., & Michael, J. (2006). *One-on-one Tutoring by Humans and Machines*. Mahwah: Erlbaum.
- Gertner, A. S., & VanLehn, K. (2000). Andes: A coached problem solving environment for physics. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Intelligent tutoring systems: 5th international Conference, ITS 2000* (pp. 133–142). New York: Springer.
- Gertner, A., Conati, C., VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. *Proceedings of the 15th national Conference on Artificial Intelligence*.
- Martin, J., & VanLehn, K. (1995). Student assessment using Bayesian nets. *International Journal of Human-Computer Studies, 42*, 575–591.
- Muldner, K., Burleson, W., van de Sande, B., & VanLehn, K. (2011). An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impacts. *User Modeling and User-Adapted Interaction, 21*(1–2), 99–135.
- Nwaigwe, A., Koedinger, K. R., VanLehn, K., Hausmann, R. G. M., & Weinstein, A. (2007). Exploring alternative methods for error attribution in learning curve analysis in intelligent tutoring systems. In R. Luckin, K. R. Koedinger, & J. Greer (Eds.), *Artificial Intelligence in Education* (pp. 246–253). Berlin: Springer.
- Ranganathan, R., VanLehn, K., & van de Sande, B. (2014). What do students do when using a step-based tutoring system? *Research and Practice in Technology Enhanced Learning, 9*(2), 323–347.

- Roll, I., Alevan, V., McLaren, B., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction, 21*, 267–280.
- Schulze, K. G., Shelby, R. H., Treacy, D. J., Wintersgill, M. C., VanLehn, K., & Gertner, A. (2000a). *Andes: An active learning intelligent tutoring system for Newtonian physics THEMES in education* (Vol. 1:2, pp. 115–136). Athens: Leader Books.
- Schulze, K. G., Shelby, R. N., Treacy, D. J., Wintersgill, M. C., VanLehn, K., Gertner, A. (2000b). Andes: An intelligent tutor for classical physics. *The Journal of Electronic Publishing, 6*(1).
- Shapiro, J. A. (2005). Algebra subsystem for an intelligent tutoring system. *International Journal of Artificial Intelligence in Education, 15*(3), 205–228.
- Siler, S., Rose, C. P., Frost, T., VanLehn, K., Koehler, P. (2002). *Evaluating knowledge construction dialogues (KCDs) versus minilesson within Andes2 and alone*. Paper presented at the Workshop on dialogue-based tutoring at ITS 2002, Biarritz, France.
- Singley, M. K., & Anderson, J. R. (1989). *The Transfer of Cognitive Skill*. Cambridge: Harvard University Press.
- van de Sande, B. (2013). Applying three models of learning to individual student log data. In D'Mello, S. K., Calvo, R. A., Olney, A. (Eds.), *Proceedings of the 6th International Conference on Educational Data Mining* (pp. 193–199): International Educational Data Mining Society.
- VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist, 46*(4), 197–221.
- VanLehn, K., & Martin, J. (1998). Evaluation of an assessment system based on Bayesian student modeling. *International Journal of Artificial Intelligence in Education, 8*(2), 179–221.
- VanLehn, K., & Zhendong, N. (2001). Bayesian student modeling, user interfaces and feedback: A sensitivity analysis. *International Journal of Artificial Intelligence in Education, 12*(2), 154–184.
- VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., Taylor, L., & Wintersgill, M. C. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education, 15*(3), 147–204.
- VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science, 31*(1), 3–62.
- Weerasinghe, A., & Mitrovic, A. (2006). Facilitating deep learning through self-explanation in an open-ended domain. *International Journal of Knowledge Based and Intelligent Engineering Systems, 10*, 3–19.