XTutor: an Intelligent Tutor System for Science and Math Based on Excel

Roxana Gheorghiu and Kurt VanLehn

Computer Science Department, University of Pittsburgh
{roxana,vanlehn}@cs.pitt.edu

Abstract. VanLehn argued that an essential feature of many intelligent tutoring systems (ITSs) is that they provide feedback and hints on every step of a multi-step solution.But if step-based feedback and hints *alone* suffice for strong learning gains, as Anderson et al. conjecture ([1]), then perhaps a lightweight tutoring system that employ *only* feedback and bottom-out hints would have advantages. This motivates the current project.Using Excel there are some immediately advantages that can be obtained: most people is familiar with its user interface and its notation for mathematical expressions, Excel already contains facilities for solving some systems of equations and it can be easy combined with many other pieces of software, making it easier for instructors to include the tutor in their course activities. Finally, web-based delivery is simple because most students already have and use Excel.

1 Introduction

It was already tested on some well-kown ITS that providing feedback and hints on every step of a multi-step solution is an essential feature. It is not clear yet whether step-based feedback and hints alone suffice for strong learning gains, as Anderson at al. conjecture ([1]).

We wonder if the spreadsheet program Excel would allow easy construction of an ITS that would teach the same class of equation-based task domains as Andes (VanLehn at al. [3]) and Pyrenees (VanLehn et al. [2]). In these task domains, students solve a problem by defining variables, writing equations and finally solving the system of equations algebraically.

2 Using XTutor

As Fig. 1 illustrates, a problem in our Excel-based tutor, XTutor, is presented as an Excel worksheet with the statement of the problem on the first row of the sheet. Rows three and four briefly explain what should be entered in each column and how. All the other rows are filled in by the student.

Figure 1 shows the worksheet after it has been filled out by the student. Each row has:

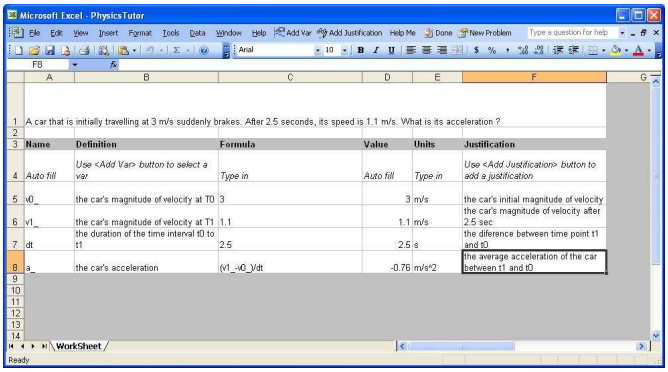

Fig. 1. Example of a problem in XTutor

1. a variable name that is automatically filled in by the tutor when the student selects its definition
2. a definition for the variable that is selected from a problem-specific menu of all possible variables definitions.
3. an algebraic formula, typed in by the student, to compute the variables value
4. a numerical value for the variable, filled in by the tutor, as the result of the previously introduced formula
5. units for the value, typed in by the student
6. the name of the principle that justifies the algebraic expression used for the variables value or other type of short justification selected by the student from a list of all possible justifications defined for that particular problem.

XTutor provides feedback on each student entry. Variables used in a formula that were not previously defined are signaled by a *#Name* type of error in the *Value* field. Algebraic expressions, units and justifications are also given immediate feedback. Moreover, if the student asks for a hint on a cell, the system can give a sequence of hints, where the final bottom out hint says exactly what should be entered in the cell.

At the end, the $< Done >$ button checks if the problem was correctly and completely solved.

3 Authoring Domain Knowledge for XTutor

XTutor is an example-tracing type of tutor. That is, the domain knowledge of the system consists of a set of examples, one per problem. The example contains at least one and possibly several complete solutions to the problem.

The author enters an example exactly as a student would when solving the problem, except that the author must enter all the cells by herself. This suffices for giving immediate feedback and bottom-out hints. If the author wants any hints besides the bottom out hint on a cell, then the hints must be entered for that cell specifically.

4 Positive And Negative Features of XTutor

The main advantage of XTutor is its simplicity, for both students and authors. It takes only minutes to learn how to use it. A second feature is that once students have seen how to use Excel in this fashion to solve problems, they can continue to do so without the help of XTutor. Thus, we expect significant transfer from XTutor to unsupported problem solving using Excel.

However, there are certainly drawbacks to XTutor. In a solution, every variable appears in two or more equations. It is sometimes not clear to the student which equation should be written in the row for a variable. Students often prefer to work forwards from the given values. Usually, this eliminates all choices. However, even using this strategy, choices sometimes remain.

Take, for example, the following problem: *How many liters of 70% alcohol solution must be added to 50 liters of 40% alcohol solution to produce 50% alcohol solution?* Suppose the student is working on a row for tot_mix, which represents quantity, in liters, of the mixture. There are two choices: *tot_mix=tot_low+tot_high* or *tot_mix=alc_mix/pr_mix*, where *tot_low* and *tot_high* are the total quantity of low and high concentration solutions, *alc_mix* is the total quantity, in liters, of alcohol in the mixture solution and *pr_mix* is the percentage of alcohol in the mixture solution. If the student picks the first version then she must first think what formula to assign to *tot_high* (or *tot_low*) since *tot_high=tot_mix-tot_low* is not permitted anymore. Similar problems appear if the second formula is used to define *tot_mix*. When assigned to more skilled solvers, such problems may actually be beneficial, as they encourage students to plan their solutions before writing them down. Thus, we see this feature as positive but with a potential for misuse.

The domain generality of XTutor has already been tested in a preliminary way by authoring problems in three task domains: physics, math and chemistry. The next step is to try it out on students, first in pilot studies and then in comparison to paper-and-pencil and an established tutoring system, such as Andes.

References

[1] 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.

[2] VanLehn, K., Bhembe, D., Chi, M., Lynch, C., Schulze, K., Shelby, R., et al. (2004). Implicit vs. explicit learning of strategies in a non-procedural skill. In J. C. Lester, R. M. Vicari, F. Paraguaca (Eds.), Intelligent Tutoring Systems: 7th International Conference (pp. 521-530). Berlin: Spring-Verlag.

[3] VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., Taylor, L., et al.(2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence and Education, 15(3), 147-204.

[4] Mitrovic, A., Suraweera, P., Martin, B. (2006). Authoring constraint-based tutors in ASPIRE. In M. Ikeda, K. Ashley, T.-W. Chan (Eds.), Intelligent Tutoring Systems: 8th International Conference, ITS 2006 (pp. 41-50). Berlin: Springer-Verlag.