Porting an Intelligent Tutoring System across Domains

Min CHI and Kurt VANLEHN
Learning Research and Development Center,
University of Pittsburgh, PA 15260 U.S.A

Abstract. One possible approach to reducing the cost of developing an intelligent tutoring system (ITS) is to reuse the components of an existing ITS. We used this approach to develop an Andes probability tutoring system by modifying the declarative knowledge of the Andes physics tutoring system. We claim that if we cluster various educational domains into groups based on their problem-solving methods [2], then it will be more efficient to port an existing ITS to a new domain in the same cluster than to build a new ITS from scratch.

Introduction

One of the main obstacles to deploying a new ITS is the significant time and cost requirements. A diverse team of computer programmers, domain experts, and educational theorists, are needed during the entire process of development and testing. Two approaches have been taken to reduce such costs. One is to use authoring tools which provide a relatively simple development environment to reduce the programming load [1][4]. Another is to reuse shared components from preexisting systems. However, little effort has been spent on the latter despite the fact that a large number of components are shared across ITSs. One of the reasons for this is that educational research is often domain-specific. Research on effective physics instruction is often conducted independently from research on probability instruction. There is often little consensus on the effectiveness of an instructional method within a single domain let alone across domains. As a result, there are no universal standards for tutoring components and most ITSs are entirely domain specific.

However, we believe that it is more efficient in terms of time and cost to port an existing ITS to a new domain than to build a new system from scratch if the two domains share the same problem-solving method (PSM) [2]. We ported our existing Andes physics tutor to a new domain, probability, by replacing the declarative knowledge within the existing physics knowledge base with new probability principles and concepts. All of the procedural knowledge, help procedures, and interface components were retained. We will describe this process in greater detail below.

1. Porting Andes from Physics to Probability

Andes is an ITS that has helped hundreds of students improve their understanding of college physics [5]. It consists of five components: a Knowledge Base (KB), a Pedagogical Module, a Problem Solver, a mathematics package, and a GUI. The KB comprises declarative domain knowledge such as principles, concepts, and problem definitions. The Pedagogical Module comprises procedural knowledge including optimal problem solving methods and instructional algorithms for the provision of hints, which include procedures for determining how students should solve problems and algorithms determining the types of hints provided to students and their level of specificity. The Problem Solver employs inference rules to automatically derive a solution graph for each problem. Each graph represents multiple solutions to the given problem. The Mathematics Package handles algebraic simplification. The GUI permits students to solve problems in a manner similar to using a pencil and paper. These components are described in much greater details in [5].

We ported Andes to a probability domain by replacing the physics KB with declarative knowledge for probability. The new KB consists of seven concepts, 10 principles (see Table 1), and 36 problem definitions. The seven concepts were: *sample space* (Ω), *single event* (X), *event complement* (~X), *intersection* ($X \cap Y$), *union* ($X \cup Y$), *conditional event* (X|Y), and *probability* (p). For each principle, we wrote three levels of hints: *pointing*, *teaching* and *Bottom-Out*. For example: for the "Addition Theorem for two events, X and Y", the pointing hint is "*Please apply the addition theorem for two events X and Y*"; the teaching hint is "*The definition of addition theorem for two events A and B is* $P(A \cup B)=P(A)+ P(B)- P(A \cap B)$. *Please apply the addition theorem on event X and Y.*"; and the bottom out hint is "*Write the equation* $P(X \cup Y) =P(X) + P(Y) - P(X \cap Y)$".

Table 1. Major Principle of Probability

Complement Theorem	Mutually Exclusive Theorem
Addition Theorem for two events	Addition Theorem for three events
De Morgan's Theorem	Independent Events
Conditional Probability	Conditional Independent Events
Total Probability Theorem	Bayes Rule

For each of the 36 problems, the Andes' Problem Solver automatically produced a solution graph. The number of principle application required to solve a problem varied from 1 to 9. Before Andes-probability was ready for testing, we also asked one of Andes' developers to set aside one afternoon to develop the one GUI component necessary for defining the probability of an event on Andes. Altogether, it took a moderately experienced ITS developer, the first author, roughly 10 days and an experienced Andes programmer less than one day to port Andes from physics to probability . Figure 1 shows the Andes-probability GUI.

There are three primary reasons why the port was so simple. First, although probability and physics are dissimilar domains at the surface level, they share PSMs which are normally described informally as ways to solve a task [2][3]. In our case, the two domains are both principle-based domains and problem solving in them involves producing a set of equations that solve for a sought quantity. Since Andes' Problem Solver is independent from its KB, it can also be used to automatically produce solution graphs for the probability problems. Generally speaking, producing these solution graphs is one of most expensive tasks when deploying an ITS. Second, the remaining components which include the Pedagogical Module, the mathematics package and the GUI, are also domain-independent and thus can also be reused. For example, the Pedagogical Module includes the domain-independent pedagogy rules such as "variables must be defined before being used". At the same time, various Andes hints can be extracted from the declarative KB instead of being embedded in the Pedagogical module. Finally, we also reused Andes' error handlers. Andes distinguishes between two types of errors for error handling: *tiny errors* and *red errors*. *Tiny errors* are those that are likely due to lack of attention such as typos. *Red errors* are those that are due to conceptual misunderstandings or a lack of knowledge. *Tiny errors* are largely domain-independent and thus we did not need to rewrite anything for them. *Red error* handling is primarily driven by the declarative KB which we had already rewritten so no additional development was required.

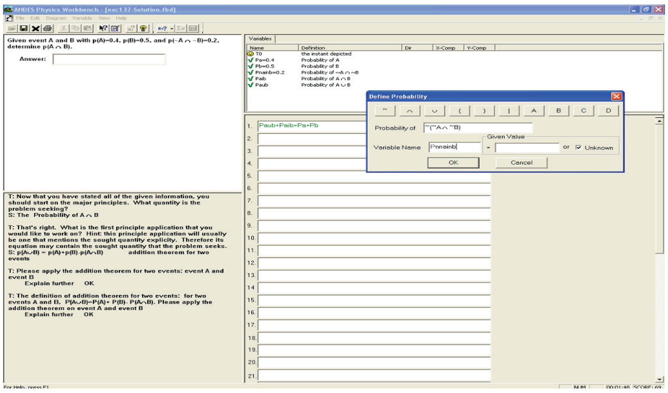

Figure 1. Andes' Interface for Probability

2. Conclusions

Developing high-level authoring system have been proven to be general and successful, e.g. the Cognitive Tutor Authoring Tools [1] and REDEEM [4], but the cost of producing them is itself very high. In this paper, we have described how porting an existing tutor to new domains might be more efficient in terms of time and cost. However, it is a less general alternative. We believe there are two essential requirements for porting to be efficient: first, the declarative KB should be constructed independently from the rest of components in the ITS; second, the two domains should share a common PSM. If either requirement is violated, we expect the cost of porting to increase radically, potentially exceeding the cost of starting from scratch.

Clearly, there are additional factors, such as cognitive similarity that will also affect the effectiveness of the porting. Those factors should be explored in future work. One way to make our approach more doable for other domains or to make this kind of technology migration easier is for the AI&Ed field to cluster various domains into groups based on common PSMs, build a digital library of ITSs, and set up standards for documentation. This research can be guided by preexisting work on the development of reusable components for knowledge-based systems [3].

3. References

[1] Aleven V, Bruce M. McLaren, Jonathan Sewall, Kenneth R. Koedinger (2006) The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains In the Proceedings of the 8th International Conference on Intelligent Tutoring Systems Jhongli, Taiwan, June 26-30, 2006

[2] Chandrasekaran, B "Towards a Taxonomy of Problem Solving Types" AI Magazine 4(1) 1983, 9-17.

[3] D. Fensel and E. Motta. Structured Development of Problem Solving Methods. IEEE Transactions on Knowledge and Data Engineering, 13(6):913–932, 2001.

[4] Major, N. 1995. REDEEM: Creating Reusable Intelligent Courseware. In Proceedings of the International Conference on AIED., Greer, J., ed., AACE, Charlottesville, VA, pp. 75-82.

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