Appeared in: In J. Greer (Ed.), Proceedings of AI-ED 95-7th World Conference on Artificial Intelligence and Education (p. 567). Charlottesville, VA: AACE.

ITS Authoring Tools: The Next Generation

Stephen B. Blessing
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213–3890
blessing+@cmu.edu

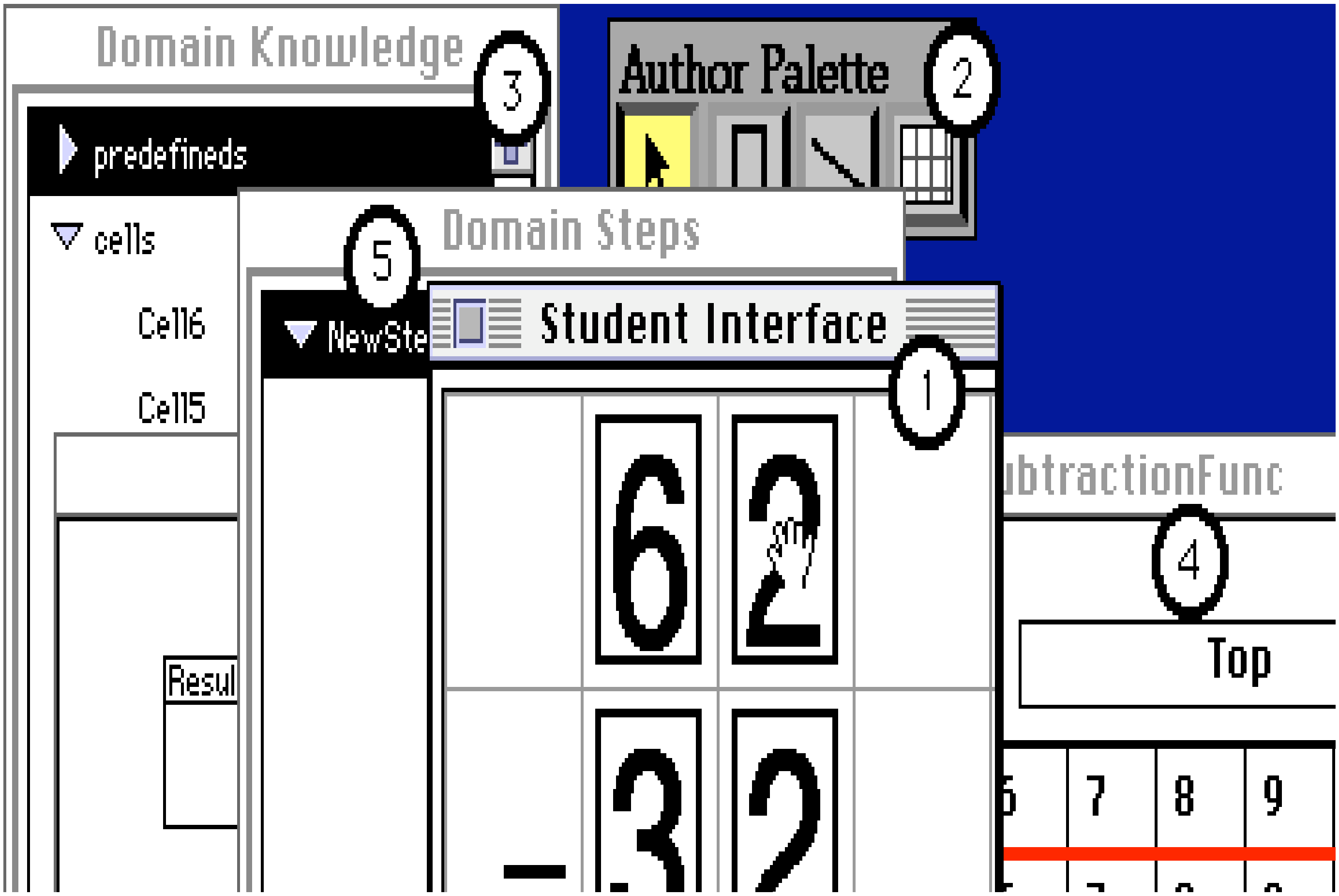
The benefits of intelligent tutoring systems (ITSs) have been shown by many researchers, with some demonstrating more than a standard deviation's improvement over traditional instruction. However, the construction of these tutors is an extremely time–intensive effort. Furthermore, creating a tutor requires not only knowledge of the tutored domain, but also knowledge of cognitive science and of programming. The goal of the present research is to drastically decrease the amount of time needed to design ITSs by reducing the amount of non–domain knowledge needed to construct an ITS.

The current research focuses on using end–user programming techniques, such as programming by demonstration (PBD), to aid in the construction of a model–tracing tutor. The figure shown below illustrates the authoring interface for a tool that creates arithmetic tutors (e.g., addition and subtraction). After constructing the interface (1), using a set of MacDraw–like tools (2), the author must define the important aspects of the problem, such as the cells that make a column (3). This corresponds to creating the appropriate working–memory elements found in many model–tracing tutors. Once this has been accomplished, the author must then make explicit the knowledge that a person implicitly uses when performing the task—the procedural knowledge. In the current tool, this is accomplished by creating "working–memory functions." These working memory functions return values based upon the interactions the author makes between them and the student interface. In the figure, a working–memory function can be seen (4) that the author can drop two numbers from the interface on, and it will return the difference of the numbers. To create the actual production rules used in the task (5), the author essentially solves problems using the student's interface, and the tool records the actions. The author can then fine–tune the productions, as well as add context–sensitive help.

It is not clear what the contribution of PBD techniques will be to future ITS authoring tools. The idea of being able to program a tutor's expert system by having the author solve problems is definitely a powerful one. However, the necessity of making each step explicit within the interface and the limits of machine learning may place bounds on the domains that can support such techniques. What may be necessary is to create multiple authoring tools, each one meeting the requirements and needs of a different domain.

This material is based upon work supported by the National Science Foundation and the Advanced Research Projects Agency under Cooperative Agreement No. CDA–940860.

Appeared in: In J. Greer (Ed.), Proceedings of AI-ED 95-7th World Conference on Artificial Intelligence and Education (p. 567). Charlottesville, VA: AACE.


Domain Knowledge
3
predefineds
cells
Cell6
Cell5
Resul
Author Palette
2
Domain Steps
5
NewSte
Student Interface
1
6
2
-
3
2
ubtractionFunc
4
Top
6
7
8
9