Story Generation to Accelerate Math Problem Authoring for Practice and Assessment

Yue Cui[1], Rohit Kumar[1], Carolyn P. Rosé[1,2], and Kenneth Koedinger[2]

[1] Language Technologies Institute and
[2] Human-Computer Interaction Institute,
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213
{ycui,rohitk,cprose,krk@cs.cmu.edu}

Abstract. We present a novel authoring infrastructure[1] for accelerating the rate of content creation for coached practice environments. In an initial authoring stage, the tool supports quick authoring of scaffolded problems, drawing upon principles established in prior intelligent tutoring authoring research. However, in contrast to earlier approaches, the result of this stage is a structured template than can be used to generate a multitude of variations on the same problem concept.

1 Morph Generation Tool

Coached problem solving environments employing some form of "model tracing" inspired by theories of problem solving from cognitive psychology have proven effective for learning, and have recently been used for developing assessment systems that can be used to do assessment during instruction. A variety of authoring environments have been developed to greatly accelerate the rate at which content for such systems can be authored [2,4]. These environments make development of coached problem solving environments easier and more cost-effective by making it possible for non-programmers to create the content. Nevertheless, authoring an item involves demonstrating alternative correct and incorrect problem solving steps and then annotating steps in the resulting representation with hint messages and feedback messages. Often it is desirable for students to work through several similar problems, which we refer to as "morphs", in the course of a year in order to perfect the relevant skills. Currently only rudimentary support for reusing the effort involved in authoring a single item is available. Research related to generating morphs has focused on generalizing the numbers, equations, or graphs that are included in a problem, sometimes in order to manipulate problem difficulty, but not the cover story [3]. We see this as an opportunity to build on earlier successes by using story generation technology to multiply the fruits of authoring effort by augmenting existing item authoring technology. Thus, in this poster we present an authoring infrastructure that was designed to templatize the output from such authoring environments so that the effort expended to author a single problem can produce multiple problems with an interesting variety of cover stories.

[1] This research was supported by a grant from the GE Foundation.

B. Woolf et al. (Eds.): ITS 2008, LNCS 5091, pp. 790–792, 2008.

The authoring process for content creation in existing environments such as the Cognitive Tutor Authoring Tools and the Assistments Builder is a two stage process in which the texts that are presented to students in the final running system are authored directly, and then these texts are embedded in the "machinery" that transforms them into fully functioning items that can be used within coached problem solving environments. In contrast, in our approach, templates are authored that can be used to instantiate a variety of texts. And a middle stage is inserted in between the initial and final stages of the original authoring process, in which the templates are instantiated in multiple ways in order to produce a variety of cover stories. From a high level, the authoring process is illustrated in Figure 1.

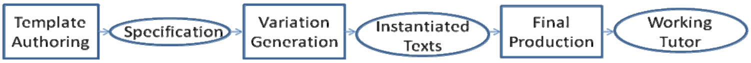


Fig. 1. Morph generation process

The morph authoring interface is illustrated in Figure 2. Notice that at the simplest level, authored texts can be templatized by inserting variables that can be instantiated in a variety of ways. The template appearing in Figure 2 could be instantiated in such a way as to produce the following two texts:

[1] "Seeing a laptop for a great discount price, Fred bought it on-line. There is a discount of 25% off the price of $700 for the laptop. How much did Fred pay after the discount?"

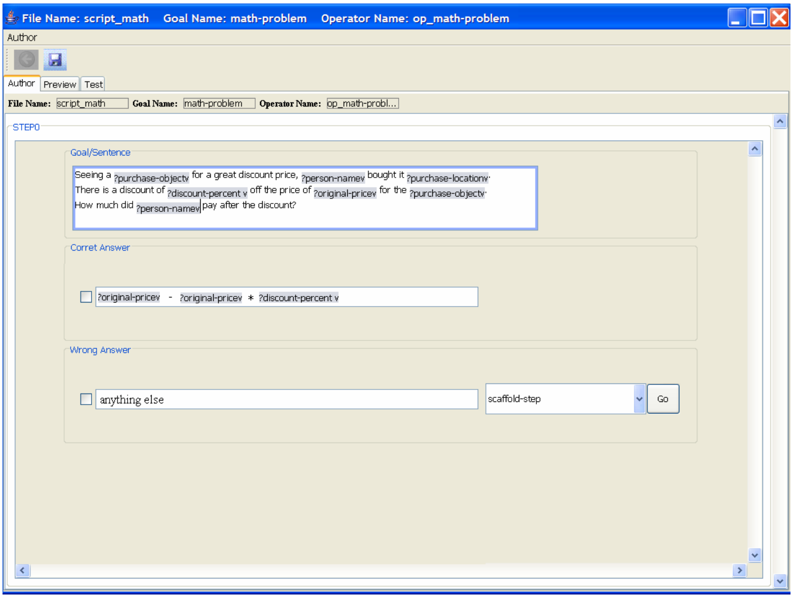
[2] "Seeing a bicycle for a great discount price, Alice bought it at Walmart. There is a discount of 10% off the price of $100 for the bicycle. How much did Alice pay after the discount?"

This is the level of variation that can be achieved with the simplest use of the authoring environment. However, the environment includes a hierarchical planner as a control structure that allows morphs to be authored in a more flexible fashion, as in the TuTalk tutorial dialogue authoring environment [1]. For example, the template in Figure 2 contains 3 sentences. Instead of authoring the template in this way, it could have been authored as 3 separate story steps. And each of those story steps could have been specified to have a range of alternative step definitions, rather than being defined directly as texts with variables. Each of those step definitions could have then been defined as texts with variables, like the text contained in Figure 2. In that way, then a great deal more variation can be achieved. For example, for the first sentence, there may be three alternative step definitions, which include the following template texts:

[1] "Seeing a ?purchase-object for a great discount price, ?person-name bought it? purchase-location."

[2]"Let's think about the discount?person-name got on a ?purchase-object? purchase-location."

[3] "?person-name wanted to buy a ?purchase-object. Buying it ?purchase-location had the advantage of an attractive discount."



File Name: script_math Goal Name: math-problem Operator Name: op_math-problem

File Name: script_math Goal Name: math-problem Operator Name: op_math-probl...

Goal/Sentence

Seeing a ?purchase-objectv for a great discount price, ?person-namev bought it ?purchase-locationv·
There is a discount of ?discount-percent v off the price of ?original-pricev for the ?purchase-objectv·
How much did ?person-namev pay after the discount?

Corret Answer

□ ?original-pricev - ?original-pricev * ?discount-percent v

Wrong Answer

□ anything else scaffold-step

Fig. 2. Morph generator interface

By defining the morph at an even higher level of abstraction, different types of problems can be generated. For example, a slight variation would allow a problem about adding on tax rather than subtracting a discount. The same alternatives for the first sentence can be reused. Furthermore, work specifying lists of possible purchase objects, names, stores, etc., can be reused across these two story types. In pilot evaluations of our morph generation approach, we have been able to generate hundreds of alternative stories from a hierarchically structured template specification.

References

1. Jordan, P., Hall, B., Ringenberg, M., Cui, Y., Rosé, C.P.: Tools for Authoring a Dialogue Agent that Participates in Learning Studies. In: Proceedings of AI in Education (2007)
2. Koedinger, K.R., Aleven, V., Heffernan, T., McLaren, B., Hockenberry, M.: Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In: Proceedings of AI in Education (2004)
3. Shute, V., Graf, E.A., Hansen, E.G.: Designing Adaptive, Diagnostic Math Assessments for Individuals With and Without Visual Disabilities. ETS Research Report (2006)
4. Turner, T.E., Macasek, M.A., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K.: The Assistment Builder: A Rapid Development Tool for ITS. In: Proceedings of ITS (2005)