WPI-CS-TR-08-03 January 24, 2008

Cost-Effective Content Creation with Variabilization

by

Manasi P. Vartak
Shane F. Almeida
Neil T. Heffernan

Computer Science Technical Report Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Cost-Effective Content Creation with Variabilization

Manasi P. Vartak, Shane F. Almeida, and Neil T. Heffernan

Worcester Polytechnic Institute
Department of Computer Science
100 Institute Road
Worcester, MA 01609
{mvartak, almeida, nth}@wpi.edu

Abstract. Traditional intelligent tutoring systems have been successful at fostering learning, but very few systems have been built due to the high cost of creation. It has been reported that it takes between 100 to 1000 hours to produce a single hour of tutoring content. Our previous research reported a reduction in the cost of authoring content through the use of pseudo-tutors, constructs that mimic cognitive tutors but are limited in scope to a single problem. Although the extreme reduction in complexity allowed teachers with no background in intelligent tutoring systems to build effective tutoring content, building multiple questions within a particular skill set required significant repetition of content. In the current work, we add some complexity back into the system by allowing teachers to generalize pseudo-tutors through the use of variables that can alter the contextual and numerical data used in the problem. We report evidence that variabilization reduces the cost of authoring similar skill problems by a factor of two. Further, this factor increases linearly with the number of instances of the problem created. We also suggest that the additional complexity is not a hindrance to teachers adopting the system and some repetition of tutoring content is acceptable to students.

Key words: authoring, variabilization

1 Introduction

Although model-tracing, rule-based tutors provide effective learning environments in a variety of domains [2], research has shown that developing a single hour of usable student content can take between 100 and 1000 hours [1, 2]. Creating cognitive tutors also requires considerable knowledge in the domains of cognitive psychology and computer science, particularly artificial intelligence rule-based programming [3]. The high ratio of creation time to usable content and the high level of expertise required have resulted in very few systems being built and consequently adopted by educators.

Authoring tools for intelligent tutoring systems is a vibrant area of research as indicated by the recent book by Murray, Blessig, and Ainsworth (2003) that reviews over a dozen authoring tools. Most of these systems involve novel ideas to speed up authoring but most of the systems have not yet "left the laboratory." One recent authoring tool, "TuTalk," has similarities to our work in that they create simple dialogues but they are more focused on features lying outside our area of interest (i.e., support in Natural Language input).

The Office of Naval Research funded the Assistment project in part to explore ways of reducing the cost of making intelligent tutoring systems. Our goal was to facilitate rapid content creation by users with little or no background in computer science and cognitive psychology. Rather than implement a rule-based tutor that generalized over a particular domain, our project focused on developing a framework and supporting tools for the creation of "pseudo-tutors" [3].

1.1 The Assistment Project

The Assistment project is a joint research effort by Worcester Polytechnic Institute and Carnegie Mellon University, and is funded by grants from the Department of Education, the National Science Foundation, and the Office of Naval Research. The mission of the Assistment project to provide cognitive-based assessment of students is supported by providing tutoring content to students, providing useful and up-to-date reports on student performance to teachers, and providing tools to allow teachers to create their own tutoring content.

Most tutoring systems are built to assess student knowledge within a set of concepts (e.g., exams) or to assist them in acquiring a certain skill (e.g., tutorials). However, the Assistment system is novel in that it accomplishes both these goals simultaneously: it assists while it assesses. The system assists students in acquiring different skills through the extensive framework of scaffolding questions, hints, and incorrect messages [4]. Assessment of student performance is provided to teachers through real-time reports based on statistical analysis and the system includes builder tools that allows teachers to create content tailored to their classes. The only requirement for using the Assistment system is registration on our website; no software needs to be obtained or installed. Our primary users are middle- and high-school teachers throughout Massachusetts who are teaching the curriculum of the Massachusetts Comprehensive Assessment System (MCAS). Presently, we have 3000 students and about fifty teachers using our system as part of their normal classes.

1.2 Pseudo-Tutors

Pseudo-tutors, a concept introduced by Koedinger et al., are constructs that mimic cognitive tutors but are limited in scope to a single problem [6]. In our system, we have implemented a simplified version of pseudo-tutors that supports only a linear progression through a problem; the pseudo-tutors in our system (i.e., a problem with any scaffolding) are called "assistments." This simplification makes content creation easier and more accessible to a general audience. Previous research has shown that our pseudo-tutor-based system can reduce the time required to build a single hour of content from 100 to 1000 hours to 10 to 30 hours [3].

Although they exhibit similar behavior to rule-based tutors, pseudo-tutors lack the ability to generalize over similar problems [3]. Pavlik et al. report that learners, particularly beginners, need practice at closely spaced intervals [9] while McCandliss and others claim that beginners benefit from practice on closely related problems [10]. Applying these results to a tutoring system requires a significant body of content addressing the same skill sets. Without the ability to generalize, separate pseudo-tutors are required for each individual problem regardless of similarities in tutoring content, something that is tedious and time consuming in the present system. Indeed, a common task among content creators in our system is "morphing:" modifying existing assistments in subtle ways (e.g., changing quantities in the problem) to create new content. Currently, about 140 morphs exist in our system out of about 5000 assistments. Our present research seeks to extend our content-building tools to facilitate the reuse of tutoring content across multiple instantiations of an item through variabilization. Our goals are to reduce per-problem costs and to determine an acceptable number of morphs for a given skill.

2 Variabilization

One of the main goals of the Assistment system was to create a tool that instructors with no experience in cognitive psychology or computer science could use to build content with ease. In order to achieve this goal, a specialized system was built around the concept of simplified pseudo-tutors. Our present research seeks to extend our content-building tools to facilitate the reuse of tutoring content across multiple instantiations of an item through the technique of variabilization. Our aim is to add some complexity to the pseudo-tutor model to increase flexibility and reduce the time required to build content while retaining the basic ease of use associated with the Assistment system.

2.1 Structure of an Assistment

At the most basic level, an assistment consists of a single main problem. For any given problem, assistance to students is available through either a series of hints or scaffolding problems. Hints are messages that provide insights and suggestions for solving a specific problem. Scaffolding problems are designed to address specific skills needed to solve the original problem. Additionally, instructive messages called "buggy messages" are provided to students if certain incorrect answers are selected. For problems without scaffolding, a student will remain in a problem until the problem is answered correctly. If scaffolding is available, the student will be programmatically advanced to the scaffolding problems in the event of an incorrect answer.

Hints, scaffolds, and buggy messages together help create assistments that are structurally simple but can address complex student behavior. The structure and the supporting interface used to build assistments is simple enough so that users with little or no computer science and cognitive psychology background can use it easily.

2.2 Implementation of Variabilization

Our current system associates variables with individual assistments. Because our model bundles the main problem, scaffold problems, answers, hints, and buggy messages together into a single assistment object, this architecture allows a broad use of variables. Each variable has a name and an associated set of possible values that relate to the context or numerical values in the problem. Depending on the degree of flexibility required, entire problem statements can be put into variables too. The number of possible values for the variables dictates the number of instantiations of an assistment that can be generated.

The assistment shown in Figure 1 addresses the Pythagorean theorem and is an assistment commonly encountered by students using our system. It has 13 hints and eight buggy messages distributed between the main problem and four scaffolds. The first step in creating a variabilized assistment is to determine possible variables in the problem. In the figure, possible candidates for variabilization, such as numerical values and certain non-essential contextual elements of the problem statement, are indicated with circles.

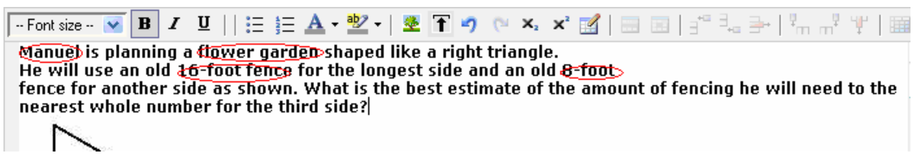

Fig. 1. The non-variabilized Pythagorean theorem Assistment with possible candidates for variables circled.

As described previously, a variable has a unique name and one or more values associated with it. A special syntax in the form of `***variable-name***` is used to refer to variables throughout the builder environment.

An important feature associated with variables is the provision to bind particular values together. This "set" provision makes it possible to use sets of values at a time rather than pick random values for variables. One use of this feature is to ensure that the quantities generated in variabilized assistments are integer values. For example, a Pythagorean theorem problem benefits from the use of Pythagorean triplets. As shown in Figure 3, the values like 3-4-5 or 10-24-26 would be chosen together. Figure 2 shows the Pythagorean assistment with variables introduced in the places identified earlier.

Generation of variables in the system is simple and follows the existing format of answers and hints. Maintaining consistency with other elements of the build

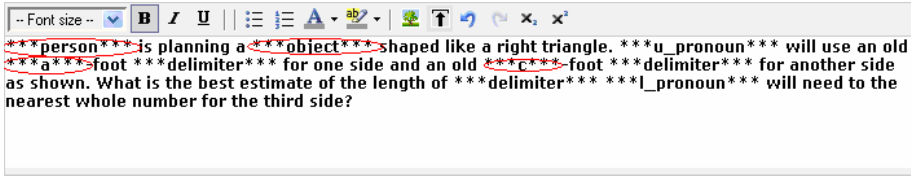

Fig. 2. A sample variabilized Assistment on the Pythagorean theorem. As shown in red, variables have be introduced for various parts of the present problem including numerical values and parts of the problem statement.

tools minimizes the learning time for content creators. A separate user-friendly widget is used for this purpose.

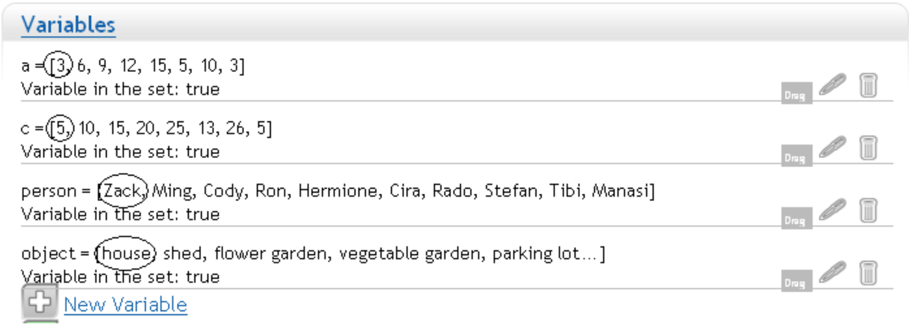

Fig. 3. Generation of variables. Each variable has a name, value and the option of having it in a "set."

After creating variables as shown above, the same `***variable-name***` syntax is used to reference them from any part of the assistment. Further, variables can be used to define functions to be used in answers, hints, and buggy messages. Figure 4 shows a function written to calculate the answer for the Pythagorean theorem problem and a different function used in the buggy message. With a similar syntax, variables and their functions can be used introduced into hints as shown in Figure 5.

This functionality allows the content creator to calculate several intermediate and final values just once and then have the system evaluate these functions for each instance of the assistment created.

Once variables have been generated and introduced into problems, scaffold questions, answers, hints, and buggy messages as required, it is possible to create multiple instances of this assistment. The main advantage of variabilization lies in the fact that once a variabilized assistment is created, new assistments including their scaffolds, answers, hints, and buggy messages can be generated instantly. The system also evaluates all defined functions for every assistment.

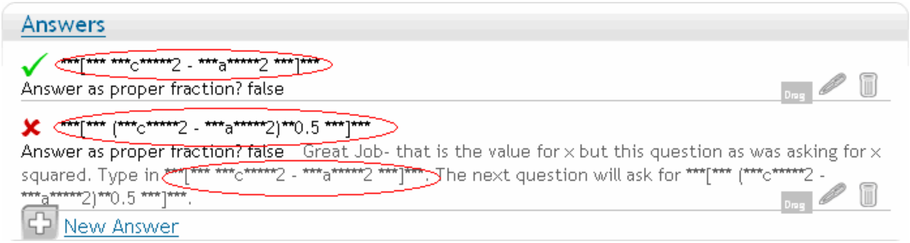

Fig. 4. Variables and functions constructed from them can be used in answers and bug messages as shown.

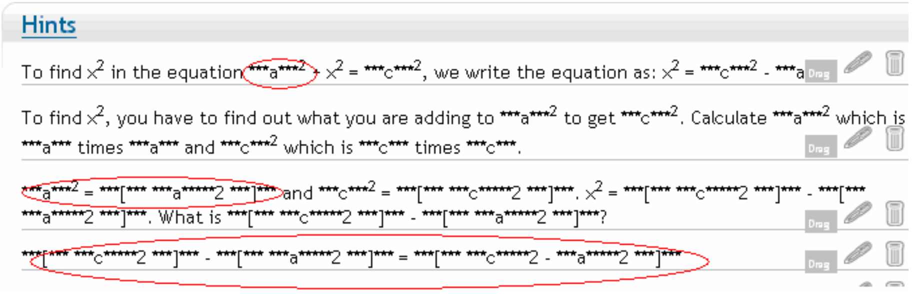

Fig. 5. Use of variables and their functions in hints.

Before creating the actual assistments, it is also possible to preview a variabilized assistment. Figure 6 shows a preview of the Pythagorean theorem assistment. In the preview, all the variables are given values and their functions are evaluated.

After the assistment has been built and previewed, multiple instances of the assistment can be created with the option "Create Variabilized Assistments." The number of generated assistments depends on the number of values specified in the sets. The Pythagorean theorem example had eight sets of variable values and hence eight different assistments are generated. Figure 7 shows the two such instantiated assistments along with a few scaffold questions, hints, answers and buggy messages. While the structure of the assistments is clearly quite similar, numeric values and non-essential contextual elements differ and provide variety.

While the Pythagorean theorem problem demonstrates the use of variables in small parts of the problem statement and in the numerical values, variables can be used to completely change the context of the problem.

3 Methods

Our study focused on the Pythagorean theorem problem discussed above since it is representative of a problem that we would wish to morph. In addition, it has four scaffold questions and 13 hints, 13 answers, and eight buggy messages, which is also representative of a typical assistment. We asked experienced content

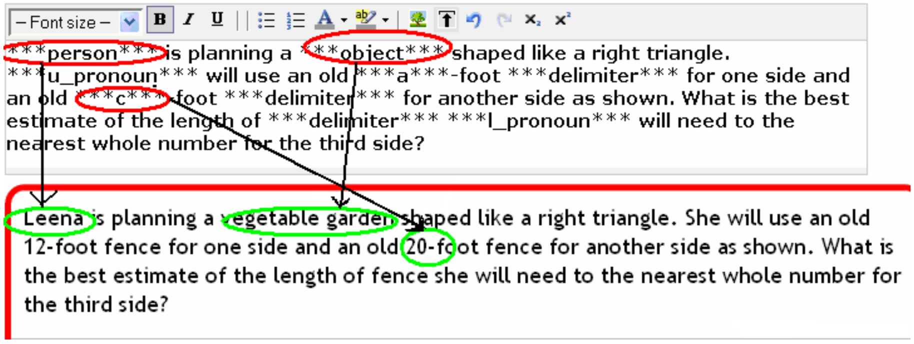

Fig. 6. Preview of variabilized Assistment. All variables and their functions have been evaluated.

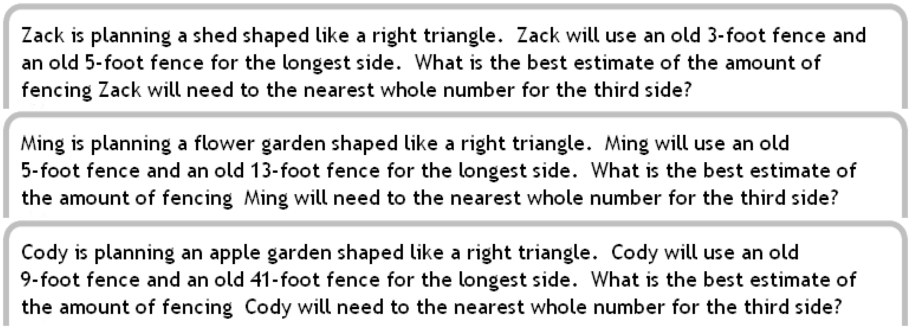

Fig. 7. Previews of the main problems of three instances of the variabilized Assistment.

creators to construct morphs of the assistment. The possible values for numeric qualities and contextual elements were provided in advance. The participants were free to use traditional morphing techniques (e.g., copy and paste). We recorded the creation time per assistment for each individual.

After the participants created their morphed assistments, we introduced the concept of variabilization with less than one hour of instruction and demonstration of the new tools. As part of the training, the participants were encouraged to experiment with variables as well as create variabilized versions of other existing assistments. At the conclusion of the training, the subjects were asked to create a single variabilized version of the Pythagorean theorem assistment that would produce the five morphs they created previously. Again, we recorded the creation time of the variabilized assistment for each individual. After a variabilized assistment was built to construct five morphs, the number of possible values for the variables was increased to create more morphs. Although the system supports more interesting variabilization, the techniques used in this study were simple and, more importantly, representative of how teachers currently morph assistments.

The five instances of the variabilized assistment were assigned to an eighth grade class consisting of 25 students. All of the students were familiar with the Assistment system and had completed a number of assignments using it in the past. The students were asked to complete an assignment consisting of the five morphed assistments but were not told the purpose of the study. At the conclusion of the assignment, each student was given a survey asking if they noticed anything about the assignment that was similar or dissimilar to other assignments they had completed in their past use of the Assistment system. If the students did notice differences, they were asked if they liked or disliked the changes. Once the initial surveys were collected, all students were told about the experiment and given a second survey. They were asked about their thoughts in general as well as whether they prefered solving similar problems as opposed to those with more variety.

Finally, the classroom instructor was asked her opinion of students completing assignments with several morphed assistments. This instructor had previously expressed interest in using morphed assistments for practice.

4 Results

Our research has implications in two areas of our system. While content creators are clearly affected by our work, there are also pedagogical implications for students.

4.1 Content Creation

Data was obtained for the development of 23 morphs of assistments created by eight individuals and 30 instances of variabilized assistments made by three participants. Table 1 lists creation times required for each of the individuals to create a morphed assistment. Our data indicates that the time to make morphs decreases with the construction of each subsequent morph, leveling off at about 12 minutes. Table 2 shows the time required to variabilize an assistment that creates the five instances that were previously created by morphing. Although the time required to create variabilized assistments decreases with each iteration, the reduction is not appreciable. After constructing a variabilized assistment that creates five morphs, it takes an average of five additional minutes to extend the assistment to generate five more.

Table 1. Average morph creation time using traditional morphing techniques

Participant	Number of Morphs	Average Time (min)
1	1	43
2	1	30
3	2	24
4	2	23
5	3	15.7
6	5	14.8
7	5	23.8
8	4	14.25
	23	20.18

Table 2. Average time required to create a variabilized assistment that produces the five previous morphs.

Participant	Number of Variabilizations	Resulting Number of Assistment	Average Time (min)	Average Time per Assistment (min)
7	2	10	36	7.2
8	2	10	38	7.6
9	2	10	41.5	8.3
	5	30	38.5	7.7

4.2 Student Acceptance

When asked about the experimental assignment, 13 out 25 students did not notice the difference between the this assignment and a typical assignment. After being told the details of the experiment, 12 students reported that they liked the similarity in the assistments, seven said that they prefered more variety in the assistments, and six indicated no clear preference. While some students commented that they appreciated the opportunity for repeated exposure to similar content, others indicated that they would have preferred having a variety of questions. Some students specifically asked for content that taught different aproaches to the same problem.

5 Analysis

Our data shows that variabilizing content reduces the average time required to make content from 20.18 minutes to 7.76 minutes. This indicates a speedup by a factor of 2.6. It is important to note that this speedup is appreciable only if more than two assistments are to be created since creating one variabilized assistment requires 38.8 minutes on average as opposed to 20.18 minutes for a morphed assistment. However, the variabilized assistment can be used to produce several instances of the assistment while the morph is essentially a single assistment. Our studies on extending a variabilized assistment indicate that creating additional instances of an assisment can be done at the rate of one instance per minute.

Tests on student acceptance indicate that approximately half the students prefer seeing similar problems as this gives them an opportunity to practice. We believe that student acceptance is closely tied with the number of similar assistments they are assigned and the degree of variability in the assistments. Considering the number of students who did not notice the content similarity, our preliminary results suggest that five to ten morphs is probably a reasonable limit for a single assignment. We expect that the relationship between number of morphs and student acceptance is roughly represented by the curve in Figure 8.

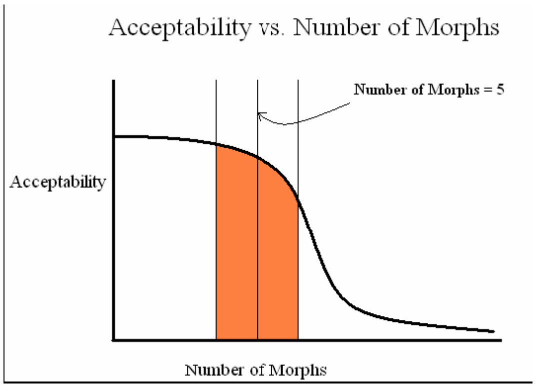

Fig. 8. The graph shows what we predict to be relationship between student acceptability of the morphs and the number of morphs.

6 Future Work

Although we preliminary results regarding student acceptability of morphs, we would like to investigate the extent to which variabilization can be utilized in our system. In particular, we are interested in determining the effectiveness of learning with variabilization versus unique content. We expect that unique content will provide greater learning, but knowing the relative worth of variabilized content compared to unique content will allow us to make pedagogical decisions in curriculum creation. In addition, variabilization will allow us to implement cognitive mastery learning, a technique in which students work on similar problems until they master a skill.

7 Conclusions

One goal of the Assistment project was to provide a system that allowed ordinary users to create and edit tutoring content. Through extreme simplification of traditional intelligent tutoring concepts, the Assistment system has made content creation available to users outside of the cognitive psychology and computer science communities. Our previous research has demonstrated that pseudo-tutor-based system and supporting build tools allow content to be created in a relatively short amount of time by ordinary users. By extending the authoring tools with variabilization, we have begun to shift slightly toward a more complex environment, but we have still retained high levels of approachability for average users. In addition, our changes have further reduced content development time by a factor of 2.6. Variabilization will allow teachers to more rapidly develop tutoring content while giving students an opportunity to practice similar problems in weaker skill areas.

8 Acknowledgements

We would like to acknowledge the Department of Education, National Science Foundation, Office of Naval Research, and the Worcester Public School System for providing the means to accomplish this research.

References

1. Anderson, J. R.: *Rules of the mind.* Hillside, NJ: Erlbaum. (1993).
2. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education* **8** 30–43 (1997).
3. Turner, T. E., Macasek, M. A., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K.: The Assitment Builder: A Rapid Development Tool for ITS. *Proceedings of ITS '05.* 929–931 (2005).
4. Razzaq, Feng, Heffernan, Koedinger, Nuzzo-Jones, Junker, Macasek, Rasmussen, Turner & Walonoski.: A Web-based authoring tool for intelligent tutors: Assessment and instructional assistance. In Nadia Nedjah, Luiza deMacedo Mourelle, Mario Neto Borges and Nival Nunesde Almeida (Eds). *Intelligent Educational Machines.* Intelligent Systems Engineering Book Series. Heidelberg, Berlin: Springer. 23–49(2007).

5. Murray, T., Blessing, S., Ainsworth, S.: *Authoring Tools for Advanced Technology Learning Environment.* Netherlands: Kluwer (2003).
6. Koedinger, K. R., Aleven, V., Heffernan, N. T., McLaren, B. & Hockenberyy, M. Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. *Proceedings of AI in Education.* 162–173 (2004).
7. Jordan, P., Hall, B., Ringenberg, M., Cui, Y., Rose, C. P. Tools for Authoring a Dialogue Agent that Participates in Learning Studies. *Proceedings of AI in Education.* 43–50 (2007).
8. Rose, C. P., Gaydos, A., Hall, B. S., Roque, A. K., VanLehn, K. Overcoming the Knowledge Engineering Bottleneck for Understanding Student Language Input. *Proceedings of AI in Education.* (2003).
9. Pavlik, P. I., & Anderson, J. R. (2005). Practice and Forgetting Effects on Vocabulary Memory: An Activation-Based Model of the Spacing Effect. *Cognitive Science.* **78(4)** 559–586 (2005).
10. McCandliss, B., Beck, I. L., Sandak, R., & Perfetti, C. (2003). Focusing attention on decoding for children with poor reading skills: Design and preliminary tests of the word building intervention. *Scientific Studies of Reading.* **7(1)** 75–104 (2003).