A La Recherche du Temps Perdu, or As Time Goes By: Where does the time go in a Reading Tutor that listens?

Jack Mostow, Greg Aist[1], Joseph Beck, Raghuvee Chalasani, Andrew Cuneo, Peng Jia, and Krishna Kadaru

Project LISTEN[2], Carnegie Mellon University
RI-NSH 4213, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890

mostow@cs.cmu.edu
http://www.cs.cmu.edu/~listen

Abstract. Analyzing the time allocation of students' activities in a school-deployed mixed initiative tutor can be illuminating but surprisingly tricky. We discuss some complementary methods that we have used to understand how tutoring time is spent, such as analyzing sample videotaped sessions by hand, and querying a database generated from session logs. We identify issues, methods, and lessons that may be relevant to other tutors. One theme is that iterative design of "non-tutoring" components can enhance a tutor's effectiveness, not by improved teaching, but by reducing the time wasted on non-learning activities. Another is that it is possible to relate student's time allocation to improvements in various outcome measures.

1 Introduction

The title of Marcel Proust's magnum opus (English translation: "In Pursuit of Lost Time") aptly expresses what this paper is about: Where does time really go when students use intelligent tutors? And how can we tell? We address both questions based on our experience with successive versions of Project LISTEN's automated Reading Tutor, which listens to children read aloud, and helps them learn to read [1].

The question of where time goes is obviously important because the effectiveness of tutorial interaction depends on how time is allocated. For example, a study of one-on-one literacy tutoring [2] compared more versus less successful tutor-student dyads, and found significant differences in their time allocation among different activities.

---

[1] Now at RIACS, NASA Ames Research Center, Moffett Field, California.

[2] This work was supported in part by the National Science Foundation under Grant No. REC-9979894. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government. We thank other members of Project LISTEN who contributed to this work, especially Al Corbett and Susan Eitelman for discussion of their video analysis; mySQL's developers; and the students and educators at the schools where the Reading Tutor records data.

Within the intelligent tutoring community, analysis of time allocation showed that Stat Lady spent much longer than necessary on remediation [5]. Followup work [6] estimated the potential reduction in tutoring time from various modifications.

In human tutoring, the tutor can control where time is spent. But when a student uses educational software, the student controls time to a greater extent – not necessarily with educational goals as the top priority. Children have their own agenda when using software, which may or may not match the desired educational outcome. As Hanna et al. [7] said, "When analyzing usage by children, we look at the goals of the product and the goals of children. The goal of the product may be to teach the alphabet, but children will probably not play with the product because they want to learn the alphabet. A child's goal may be to explore and find out what happens or to win a game." For example, children spend much of the time in "edutainment" software playing with on-screen animations [8], clicking on them a considerable amount [9]. The Reading Tutor's goal is to help students learn to read. A student's goal may be reading a particular story, writing a story, exploring the story menus, or something else.

Ideally a session on any intelligent tutor would be devoted almost entirely to educational useful activities, with a minimum of wasted time. But in reality, considerable time may be spent non-productively, whether from confusion or on purpose. What actually happens? More precisely, we want to know:

1. *What typically happened?* On average, where did time go? This question is important in identifying bottlenecks.
2. *How did students differ?* For example, did a few of the students waste most of their time? That is, where are the loopholes or escape routes that let students spend time on the tutor without useful learning?
3. *How did differences affect outcomes?* Where did high- and low-gain students differ? That is, how do different allocations of time influence – or at least reflect – what is learned by students?
4. *Did students fall into a few types?* What were the principal components of variation? Were there clusters? That is, what types of students are there? How can we identify them? To what extent do their differing profiles explain or predict what and how they learn?

This paper looks at #1 and #2, and begins to address #3, in the context of Project LISTEN's Reading Tutor.

2 Project LISTEN's Reading Tutor

Project LISTEN's Reading Tutor is shown in and described in detail elsewhere [1]. Here we summarize the aspects relevant to this paper.

Children have used the Reading Tutor at schools since 1996. A session on the Reading Tutor operates as follows. The Reading Tutor runs on dedicated PCs configured to automatically reboot themselves and launch the Reading Tutor every morning. Launching takes a minute or two, so the Reading Tutor is normally left running. First the student logs in to the Reading Tutor by clicking a *Hello* icon,

selecting her name from a menu, selecting her birth month from another menu, and reading her name aloud to make sure the Reading Tutor can hear spoken input.

A new student receives an automated tutorial on how to operate the Reading Tutor. Next it's time to pick an activity, such as a story to read or write. The 2001 Reading Tutor takes turns with the student picking activities. It picks (or encourages the student to pick) stories at a recommended reading level, which it adjusts based on the student's performance. A talking menu (see 1-click picker in Fig. 1c) lists stories to pick and provides access to more stories at the same level, and to other story levels.

The student then performs the activity with the assistance of the Reading Tutor. The Reading Tutor may insert a preview before the activity and/or a review after the activity. The activity lasts until the student completes it, clicks *Back* through each sentence or step of the activity to return to story-picking, clicks *Goodbye* to stop using the Reading Tutor, or remains inactive long enough to trigger a timeout.

Controlled comparisons of successive versions of the Reading Tutor have shown significantly higher gains than baseline instruction or independent reading practice in word comprehension [10], reading comprehension [11], and other reading skills. Although these results are encouraging, we believe that the Reading Tutor has the

potential to be even more effective. To tap that potential, we first need to analyze where the time goes when children use the Reading Tutor.

Fig. 1. a. adult-assisted picker, 1996***;*** **b.** 2-click picker, 1998; **c.** 1-click picker, 1999

3 Analyzing Time Allocation

Determining how students spend time on a tutor is difficult for several reasons:

- Multimodal dialogue is rich, and students' speech data are especially messy.
- Student behavior varies among students, and may be affected by observation.
- The tutor itself may change, both in how it appears to students and how it logs their interactions.
- Logs introduced for other purposes, such as debugging, may not be conducive to analysis.
- Anomalies, rare events, software crashes, and missing data can distort results of automated analysis methods that lack common sense.

To answer questions about time allocation in the Reading Tutor, we have therefore combined various complementary analysis methods, including live observation, anecdotal reports, video analysis, manual inspection of sample data captured by the tutor, and automated analysis of tutor-recorded event logs.

Our automated methods started with special-purpose perl scripts to parse this data to answer specific questions. More recently, we have parsed logged data from 2000-2001 into a database on which we can run SQL queries to answer a broader class of questions. It takes considerable time to parse and import into our database an entire year's worth of data down to the millisecond level, so we are working with a subset of the data for students who used the Reading Tutor for the entire year.

Automated methods permit comprehensive analysis of the copious data from months of daily use by hundreds of children. However, the mappings from actual tutoring to recorded logs to analysis results are too complex to trust on their own. There is no substitute for picking several cases randomly (to ensure a representative sample) and analyzing them by hand (to include common sense).

Because the logs are too detailed to see the forest for the trees, we have also found it important to generate more understandable views at different levels of detail. At first we extended the Reading Tutor to generate such views itself, such as class rosters and student portfolios. Now we use a database to dynamically generate human-readable views in the form of HTML tables at different levels.

We now analyze several aspects of time allocation in various versions of the Reading Tutor, including overhead, delay, mixed initiative, and distribution of time among different functions and interaction types.

4 Session Overhead: Logging In

The Reading Tutor uses login mechanisms to identify and authenticate students so that it can distinguish among them, keep their records straight, and adapt its behavior to the individual student. We have not previously quantified how long login takes. However, we observed in early versions of the Reading Tutor, in a summer lab used by over 60 children, that finding one's name on a long list is a hard task for a small child. More recent versions of the Reading Tutor display only the students who use that particular machine, and read the list aloud. Analysis of data from 2000-2001 shows that login averaged 15 seconds on a machine with 4 readers enrolled.

To reduce the risk of one child reading under another's login, the Reading Tutor times out after 15 seconds of inactivity in case the current reader has gotten up and left, so as to ensure that the next reader logs in as him- or herself. If the same reader is still there, she must log back in, which takes up additional time. How often does this occur? Analysis [11] of 541 sessions in a controlled classroom evaluation of the Spring 1998 Reading Tutor found a total of 1028 "mini-sessions" ("time from login to *Goodbye* or timeout") each lasting an average of 6.2 minutes. The difference between the total session time and the total mini-session time averaged 1.6 minutes. Apparently the average 13.5 minute session in Spring 1998 included a nearly 2-minute hiatus, followed by logging back in. Analysis of the database from 2000-2001

showed that total session length averaged 21 minutes. 38% of sessions included one or more hiatuses, whose duration averaged 1 minute and 21 seconds.

5 Response Time: Waiting for the Reading Tutor

Another source of waste is the time the student spends after processing a stimulus, waiting for the computer to respond. Response delay can eat up an embarrassingly large portion of time on tutor. Delayed response has adverse indirect consequences as well, by causing the student's attention to wander off-task, wasting additional precious seconds before coming back on-task after the tutor responds.

Previous analysis [10] of a sample of 25 sessions videotaped out of 3,833 sessions from the 1999-2000 Reading Tutor revealed that in the course of an average 20-minute session, the student spent over 8 minutes waiting for the Reading Tutor to respond – almost as long as the 9 minutes actually spent reading!

Part of this delay involved several seconds of preparatory computation (especially database accesses) prior to displaying the next sentence to the student. This component of delay was reduced by rewriting code to make it more efficient. Another source of delay involved deliberately waiting for 2 seconds of silence before responding, to ensure that the student was done speaking. This delay was reduced by modifying the Reading Tutor to respond more promptly when it heard the last two words of the sentence (at the cost of barging in prematurely a bit more often).

However, the sample was not large enough to characterize how time spent waiting *varied* among students. Worse, this sample was vulnerable to observer effects due to the presence of the video camera operator, so it was not even guaranteed to be representative of typical sessions. Finally, we wanted to determine whether wait time was still a problem now that we had sped up Reading Tutor response time.

To calculate wait time for the 2000-2001 school year, we used the database generated from log data. Communication between the Reading Tutor and the student is mixed-initiative, so it is difficult to define precisely when a turn ends. If the student finishes reading a sentence, waits for a second, becomes frustrated, rereads the sentence, and waits for a few more seconds, how much of that time should be counted as waiting for the Reading Tutor? The manual analysis used human coders' listening and judgment to distinguish between "processing the stimulus" (that is, reading the displayed sentence) versus "waiting" (such as rereading the sentence when the Reading Tutor didn't respond quickly enough). We decided to estimate wait time as starting when the Reading Tutor accepted the last word of the sentence, and ending when the Reading Tutor displayed the next sentence, or reached the end of the passage. The resulting estimate should provide an upper bound in that it overestimates delay time compared to the videotape analysis.

This estimate indicated that students averaged less than 16% of total session time waiting in stories, and less than 4% waiting in previews and reviews. According to this analysis, wait time per 20-minute session averaged less than half the 8-minute average seen in the 1999-2000 Reading Tutor, according to hand-coded videos. Apparently the changes to reduce wait time had succeeded.

6 Task Choice Overhead: Picking a Story to Read

What about the time it takes to choose a story? Previous work [12] compared the time to pick stories in 3 versions of the Reading Tutor, shown in Fig. 1. To include time spent "browsing," the comparison measured the time to settle on a story as the time from the last sentence of one story to the first sentence of the next story that the student stayed in long enough to proceed to the second sentence.

To estimate story selection time in the 1996 through 1999 versions, 10 transitions were chosen at random and analyzed by hand [3]. This procedure exposed behaviors (e.g. reading the same story twice in a row) and anomalies (e.g. timing out) that a more automated analysis might not have found.

In the 1996 version, children used the Reading Tutor under the individual supervision of a school aide who helped them pick stories using a generic Windows file picker. This process averaged about 3 minutes – a considerable time cost for each story choice, especially given that sessions averaged only 14 minutes in duration [13].

In the 1998 version, children used the Reading Tutor in their regular classrooms. The student clicked on a story title to make the Reading Tutor speak it, and clicked on an *OK* button to pick that story. Including time spent rating the just-completed story as "easy, hard, or just right" and "fun, boring, scary, or silly," this process took about 2 minutes – about as long as it took to actually read the story, according to analysis of log data for the 1931 story readings in all 541 sessions in a spring 1998 study [11]. 1931/541 comes to an average of 3.6 stories per session. So roughly half the average session duration of 13.5 minutes was spent choosing stories!

In the 1999 version, the Reading Tutor recited the menu, and a single click sufficed to pick the story. Moreover, the Reading Tutor took turns picking stories. Videotape analysis showed the time to choose each story averaged only about 30 seconds – significantly shorter ($p = .02$).

To confirm the reduction in story selection time, we queried the database of logs from the 2000-2001 Reading Tutor, using the same criterion as before, but averaging per student. Student story choice averaged 33 seconds, ranging from 16 seconds for the fastest student to 50 seconds for the slowest. Reading Tutor choice averaged 5 seconds, ranging from 4 seconds for the fastest student to 11 seconds for the slowest.

To summarize, in 1996, story choice took about 3 minutes using a generic file picker. In 1998 it required about 2 minutes using a select-and-confirm picker. In 1999 it required about 30 seconds using a one-click picker. In 2000 it averaged only 19 seconds, based on automated analysis of hundreds of sessions.

7 Time Battles: Equal Time vs. Equal Turns

Did turn taking work as designed? That is, did the Reading Tutor manage to equalize the number of stories chosen by student and tutor? One source of evidence is an "opinion poll" story review activity in the 2000-2001 Reading Tutor that asked the reader to rate the quality, difficulty, and length of the story. Analysis of the poll data showed that of the 4,560 completed story readings represented in the poll, 2,296 were

chosen by the student, and 2,264 by the Reading Tutor. Thus the turn taking mechanism successfully achieved 50-50 division in terms of *finishing* stories.

As for time allocation, querying our database showed that students spent 70% of their story time on stories selected by the Reading Tutor and only 30% of their time on stories that they picked. The reason is that students generally picked stories less challenging than the Reading Tutor's selections, and students were likelier to back out of a story the Reading Tutor picked (which did not affect whose turn it was to pick).

8 Where Time Went in the 2000-2001 Reading Tutor

Table 1 summarizes where time went for 34 students whose 2000-2001 data we have parsed so far into the database. To avoid skewing the results toward students who used the Reading Tutor more, we first computed per-student averages, and then averaged across students. "Story" refers to time spent reading stories, writing, listening to the tutor read, and waiting for the tutor to respond. "Pre/Review" is time spent on activities that precede and/or follow a story, such as introducing or practicing new words. "Tutorial" is time spent on three tutorials that train students how to operate the Reading Tutor. "Task choice" is time selecting stories. "Out of tutor" is time spent logging in to the Reading Tutor, and time spent outside of a session because the student logged out, the Reading Tutor timed out due to student inactivity, or the computer crashed.

Table 1. Per-student breakdown of time spent on Reading Tutor

	Mean	Min	Max
Stories	67%	43%	80%
Pre/Review	13%	6%	46%
Tutorial	2%	0%	9%
Task choice	5%	1%	13%
Out of tutor	7%	2%	14%

But what did student and tutor actually do during these educational activities? Anecdotal field reports brought to our attention one student who managed to spend almost all his time on the Reading Tutor "writing" junk stories. How did he defeat the intent of the turn taking mechanism? Quite simply: When it was the Reading Tutor's turn to pick a story, he read the story. The Reading Tutor was picking stories that he read in a couple of minutes – not quite fast enough to get promoted to higher levels. But when it was his turn, he picked a writing activity and then often spent the rest of the session in the story editor. With this student, we were winning the story choice battle, but losing the time on task war. Changing his behavior took a visit by his mother to the lab, after which he followed her orders to read instead of "write."

The Reading Tutor's stories, previews, and reviews are all expressed as sequences of a few types of steps. The most common types are assisted reading and writing, both of which may range from single words to sentences to entire stories. Other types of steps include narrating student-written text, listening to the Reading Tutor read

aloud, choosing from a talking menu, and oral spelling. Fig. 2 shows how time spent on stories and pre/reviews was split among these types of steps.

In general, few students spent a significant portion of their time writing. In our database of 34 students, the minimum was 2%, the average was 10%, and the maximum was 43%. Moreover, this analysis includes all writing – not just writing stories, but exercises such as typing in a "trouble word" to review it.

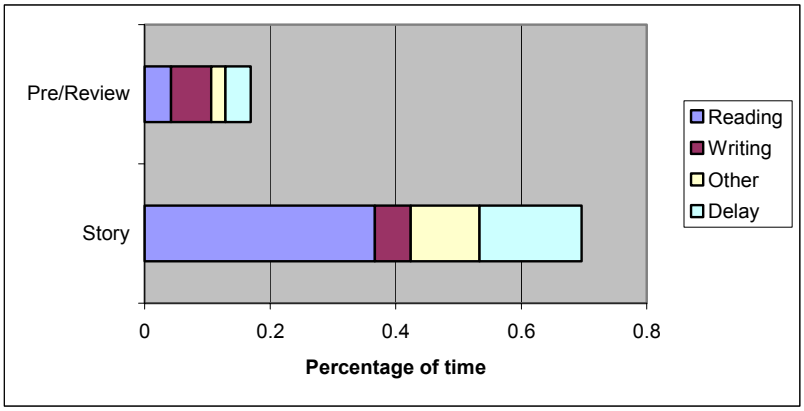

Fig. 2. Where the time went (averaged per student)

9 How Did Time Allocation Relate to Student Gains?

To compute how students' reading abilities improved over the course of a year, we pre- and post-tested specific reading skills using 4 subtests of the Woodcock Reading Mastery Test (WRMT) [14], an individually administered reading test. To control for pretest scores, we computed the gains (i.e. the posttest – pretest score for each student).

We used partial correlations in SPSS to relate students' gains on these tests to the percentages of time they spent reading, writing, in Pre/Review activities, picking stories, or in hiatus between parts of a session. To avoid confounds with initial differences among students, we controlled for student grade and (grade-normed) pretest score. Percentage of time spent on Pre/Review activities correlated positively with gains in word identification ($R=.40$, $p=0.03$) and word comprehension ($R=.51$, $p=.004$). Percentage of time spent writing correlated negatively with gains in word attack ($R=-.36$, $p=0.05$) and word identification ($R=-.38$, $p=0.03$).

These results suggest that the Pre/Review activities succeeded in helping students learn new words, since the percentage of time spent on those activities accounted for variance that grade and pretest did not. Time spent "writing" at the expense of reading may have hurt students who needed to improve their word decoding skills. Also, time spent outside of tutoring activities apparently did little harm, since the percentage of time spent picking stories or outside of the Reading Tutor did not correlate negatively with gains. However, we have not proved that time allocation

actually *caused* gains; perhaps it only *reflected* gains, or individual differences (such as motivation) that our analysis did not control for.

10 Conclusions

This paper addressed the question of where tutoring time goes, and how to find out. More specifically, how much tutoring time goes to educational activities, as opposed to overhead? How is time spent during those activities? How much time is wasted waiting for the tutor to respond? How is time allocation related to student gains? We addressed each of these questions in the context of the Reading Tutor, in some cases quantifying improvements between successive versions. We identified several methods for answering these questions, each with its own strengths and weaknesses.

Analyzing videotaped sessions is informative but too labor-intensive for more than a few sessions. Videotaping provides a detailed record, but may distort student behavior. Manual analysis allows human judgment, but hence is harder to replicate.

Automated analysis of tutor logs is more comprehensive but less trustworthy. It can compute large-sample statistics, but lacks common sense. Logs of multimodal tutorial dialogue designed for other purposes, such as debugging and performance speedup, are messy and bug-prone to parse and analyze. Log formats tend to evolve along with the tutor, limiting parsing and analysis code to specific versions of the tutor. Instantaneous events may be easy to log on a "fire and forget" basis, but events that span time intervals are trickier to log correctly because their interim state must be stored, and the tutor must remember to log their completion in every possible case. The log format is designed before the logs are generated, and therefore may not anticipate needs that do not become clear until data analysis is underway.

Organizing log data into a database incurs a considerable up-front cost to design the database. Also, populating the database with parsed logs takes time – too much to redo every time a bug is found. That said, SQL queries have been a much quicker way of answering research questions than writing specialized analysis scripts in perl to operate directly on logs as in [1]. We also consider the queries somewhat trustworthier because they are shorter and subject to fewer kinds of programming errors than general-purpose procedural code. However, writing SQL queries to answer research questions is a difficult art to master, and is no guarantee of correctness. One simple but useful lesson when computing averages is also to compute minimum and maximum, as a quick way to spotlight anomalous cases. For example, a supposed 12-minute-long wait exposed a missing case in our analysis.

Analyzing randomly chosen examples by hand applies to many more research problems than time allocation in tutors, but it is very helpful in getting a sense of typical student-tutor interaction, and in spotting bugs. At first such analysis consisted of consulting the detailed log, or the speech recognizer output for individual utterances. The detail made it hard to see the forest for the trees. Then we augmented the Reading Tutor to output student portfolios listing each session's activities [10]. Now we have a log viewer that uses the database to dynamically generate human-readable views at different grain sizes. Such views are invaluable in exposing previously hidden bugs, cases, and patterns.

Like Proust, we have focused on the "pursuit of lost time." How can analyzing where time goes improve tutors? Identifying bottlenecks can help reduce time waste. Relating time allocation to student gains indicates which activities seem to help which skills. We hope that the issues, methods, and lessons identified in this paper help researchers make time allocation more efficient in other automated tutors.

References

1. Mostow, J. and G. Aist. Evaluating tutors that listen: An overview of Project LISTEN. In *Smart Machines in Education*, K. Forbus and P. Feltovich, Editors. 2001, MIT/AAAI Press.
2. Juel, C. What makes literacy tutoring effective? *Reading Research Quarterly*, 1996. *31*(3): p. 268-289.
3. Aist, G. and J. Mostow. Faster, better task choice in a reading tutor that listens. In *Speech Technology for Language Learning*, P. DeCloque and M. Holland, Editors. in press, Swets & Zeitlinger Publishers: The Netherlands.
4. Aist, G. Challenges for a mixed initiative spoken dialog system for oral reading tutoring. *Proc. Computational Models for Mixed Initiative Interaction: Working Notes of the AAAI 1997 Spring Symposium*. 1997.
5. Shute, V. SMART Evaluation: Cognitive Diagnosis, Mastery Learning and Remediation. *Proc. 7th World Conference on Artificial Intelligence in Education*. 1995. Washington, DC: Springer-Verlag.
6. Gluck, K.A., V.J. Shute, J.R. Anderson, and M.C. Lovett. Deconstructing a Computer-Based Tutor: Striving for Better Learning Efficiency in Stat Lady. *Proc. 4th International Conference on Intelligent Tutoring Systems*. 1998. San Antonio, Texas: Springer-Verlag.
7. Hanna, L., K. Risden, M. Czerwinski, and K.J. Alexander. The role of usability research in designing children's computer products. In *The Design of Children's Technology*, A. Druin, Editor. 1999, Morgan Kaufmann: San Fransisco. p. 3-26.
8. Snow, C.E., M.S. Burns, and P. Griffin. Preventing Reading Difficulties in Young Children. 1998, National Academy Press: Washington D.C.
9. Underwood, G. and J.D.M. Underwood. Children's interactions and learning outcomes with interactive talking books. *Computers in Education*, 1998. *30*(1/2): p. 95-102.
10. Mostow, J., G. Aist, P. Burkhead, A. Corbett, A. Cuneo, S. Eitelman, C. Huang, B. Junker, C. Platz, M.B. Sklar, and B. Tobin. A controlled evaluation of computer- versus human-assisted oral reading. In *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future*, J.D. Moore, C.L. Redfield, and W.L. Johnson, Editors. 2001, Amsterdam: IOS Press: San Antonio, Texas. p. 586-588.
11. Mostow, J., G. Aist, C. Huang, B. Junker, R. Kennedy, H. Lan, D.L. IV, R. O'Connor, R. Tassone, B. Tobin, and A. Wierman. 4-Month Evaluation of a Learner-controlled Reading Tutor that Listens. In *Speech Technology for Language Learning*, P. DeCloque and M. Holland, Editors. in press, Swets & Zeitlinger Publishers: The Netherlands.
12. Aist, G. and J. Mostow. Improving story choice in a reading tutor that listens. *Proc. Fifth International Conference on Intelligent Tutoring Systems (ITS'2000)*. 2000. Montreal, Canada.
13. Aist, G. and J. Mostow. When Speech Input is Not an Afterthought: A Reading Tutor that Listens. *Proc. Workshop on Perceptual User Interfaces*. 1997. Banff, Canada.
14. Woodcock, R.W. *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU)*. 1998, Circle Pines, Minnesota: American Guidance Service.