A Non-Linear Approach to Curriculum Design: The Role of Behavior Analysis in Building an Effective Reading Program

Janet S. Twyman, Ph.D.
T. V. Joe Layng, Ph.D.
Greg Stikeleather, MA.
Kelly A. Hobbins, MA.

2 Vice President Instructional Development, Headsprout, 127 Broadway Ave. E., Suite 300, Seattle, WA 98102; email janet@headsprout.com
A Non-Linear Approach to Curriculum Design: The Role of Behavior Analysis in Building an Effective Reading Program

Janet S. Twyman, T. V. Joe Layng, Greg Stikeleather, and Kelly A. Hobbins
Headsprout, Seattle, WA
www.headsprout.com

Whether or not teaching reading is rocket science may be debated, but clearly, teaching it effectively, efficiently, and across a variety of learners is learning science. In this congressionally recognized “Decade of Behavior” it is becoming increasingly clear that if American education is going to improve, we will have to begin applying principles and procedures derived from the learning sciences directly to teaching. This trend toward recognizing the critical importance of a scientifically informed approach to teaching is indicated by the new federal requirements for scientifically-based instruction and in such statements as the U.S. Department of Education’s (1993) publication: Toward a New Science of Instruction: Programmatic Investigations in Cognitive Science and Education.

Headsprout, an Internet-based learning sciences company, has not only applied the principles derived from the scientific study of learning to the teaching of fundamental reading skills, but has turned the process of building such a program into a science itself, with valid, empirical, replicable results. But what do we mean by “turned the process of building such a program into a science itself?” What do we mean by “science?”

Science is not simply a collection of facts, observations, interpretations, or theories (Brownowski, 1956). The process by which we develop and test our interpretations or theories is science. Science is a process of searching for fundamental and universal principles that can offer parsimonious accounts of how observed changes in one set of conditions can result in changes in another set of conditions. The process involves building, testing, and connecting falsifiable accounts to describe, explain and often predict. The method includes inferences, repeatable experiments and observations that select one set of inferences over others, and the positing of new inferences. The prime criterion in determining the usefulness of an inference is the extent to which the inference correctly makes predictions or explains phenomena verifiable by independent observers, and stands tests of falsifiability. Specifically:

Inferences must be falsifiable:

There must be a way to prove the inference wrong. If we can't prove it wrong, it is not a scientific theory. This idea of an inference being falsifiable is one of the most important aspects of science.

Inferences must be able to predict:

All scientific inferences must have some predictive value. Even if an inference itself does not make predictions, it does have consequences and can be used to make some sort of predictions.
Inferences must be economical:

It is important that an inference of set, be able to provide as complete an account as is possible for the observed relations, with the simplest set of principles or axioms possible – in other words, achieve parsimony.

Inferences must be replicable:

It is not acceptable that only one person, or only one group can obtain results that support the inference. Anyone using proper procedures must be able to achieve the same results.

Inferences must engender confidence:

We have degrees of confidence in inferences, sometimes very strong, but none is absolute. The more an inference has been used successfully in the past, that is, remained intact after repeated attempts at falsification, and the more it seems to fit in with other inferences and observations, the more confidence scientists have in the inference.

Accordingly, there can be a scientific method to designing good instruction. Inferences about what and how to teach are developed from the current knowledge base. Instructional procedures are designed, tested, evaluated, revised and retested. Falsified, or ineffective strategies are modified or thrown out. Patterns of behavior change are identified and connected. And finally the instructional methods are verified across a variety of individual learners in different contexts, even across time and space.

Headsprout’s inferences about how to build a reading repertoire fit within the scientific process. Data from over 13 million responses, across over 6,000 learners has gone into Headsprout’s scientific approach to designing its early reading program. Applying a single-subject experimental design (Johnston & Pennypacker, 1993; Neuman & McCormick, 1995; 2002; Sidman, 1960), some of our inferences were falsified, while others are still supported by data. These data allow us to predict other behavior-environment relations, and replicate them over the project’s large scale. However, one cannot take those data points and follow a straight path from where we began to our current design. Our approach was (and still is) nonlinear, circuitous, and replete with continuous evaluation and revision.

Nonlinear Instructional Design

We borrow the term “nonlinear” from Izzy Goldiamond (1975; 1979; 1984). We use it, as Goldiamond did, to refer to occasions where the “behavior of interest” (the target behavior, the terminal behavior, etc.) is not simply a function of the occasion and consequences (and their history) that immediately surround it, but of the occasions and consequences of alternative patterns as well (and their history). The likelihood of saying “yes” is a function not only of the past consequences of yes, but of the past consequences of saying “no.” A linear analysis looks at the current and past consequences of the “yes” response. For example, saying “yes” to a wedding
invitation might be the result of one’s history of having fun at weddings, a free meal, or seeing Uncle Leo’s outlandish behavior at family events. However saying, “yes” to attending a less than favorite cousin’s wedding may have less to do with the consequences occasioned by a “yes” response and everything to do with the family uproar over a “no” response. It may not be the case that the presently observed consequences exert the greatest control over the behavior of interest, but instead the consequences of available alternative behavior. As we will see, such non-linearity extends to all elements of the consequential and their relation to each other and other variables that might establish or potentiate the likelihood of their participation as an element in the contingency.

The Programming Process

One typical approach to instructional design is to apply a top-down process. The goal is identified then broken down into smaller steps, then checked for “social agreement” (Do experts, or at least those with some familiarity think it’s reasonable?) and then finally placed in front of students. “Good” designers will take data and note if their students fail, and redo some portion of the program. Often this may be done in the context of field-testing with groups of learners. If the overall group tends to meet the goal, again often determined by consensus and learner emotional reaction, then the product is considered finished. Other designers, and often many curriculum publishers, fail to do the last two steps (test and revise). Such programs may “present” content that is derived from scientific principles, but the program itself may not the criteria of a scientifically developed program.

In 1967 Sue Markle and Phil Tiemann described their instructional programming process and noted that the entire instructional design process determines whether or not an instructional product will fulfill its vision. Markle and Tiemann’s (1967) process is a scientific control–analysis system that is both logical and necessary, yet it is seldom followed (Cook, 1983; Markle, 1969; 1991). One factor that may have contributed to its infrequent adoption may be that there are few examples of its application on a large scale that can serve as a guide to others who are interested in producing quality instructional materials. Another, perhaps greater obstacle, is the time and expertise required to fully implement all elements of the process.

Markle and Tiemann’s programming process can be slightly updated and summarized as follows (See Figure 1.):

1. Perform a content analysis
2. State the objectives
3. Determine the criterion tests
4. Establish the required entry behavior
5. Build the instructional sequence
6. Use performance data to continually adjust the instructional sequence (5) until it meets the objectives (2)
7. Build in maintaining consequences, an additional step in the process that was added from Goldiamond (1974).
The learner begins at 4 (Entry Repertoire) goes to 5 (Instructional Sequence) is evaluated at 3 (Criterion Tests) to determine if 2 (Instructional Objectives) has been reached. This sounds very much like how instructional designers or teachers design instruction. However, in the Markle and Tiemann process, the student does not progress through a sequence the same way as the program was built. Herein lies one of the key differences that define the approach taken here.

Markle and Tiemann’s strategy is a programming process approach. It is built around a scientific, control–analysis strategy. Accordingly, the process used in building the program is critical to instructional success.

How did Headsprout apply this nonlinear, programming process? To illustrate the design process, we will go through the progression, then we’ll describe the revision cycles, the nonlinear analysis and phases of empirical testing.

**Step 1. Content Analysis.**
First we decided what our program would be about: Program design usually begins with at least a general statement of goals. Goals are often accomplishment based and are general statements of what is to be achieved (Mager, 1997). Our goal was to:

“Teach reading/decoding skills at the mid 2nd grade level in 24 instructional hours.”

Based on the goal, the content is analyzed. The types of learning (kinesthetic, simple cognitive, complex cognitive) (Tiemann & Markle, 1991), the hierarchy of skills, and the relation of one skill set to another are determined. From this analysis the final objectives and the criterion tests for each objective are derived (See Figure 2.)
We studied the existing literature base. We did a thorough analysis of early reading skills, what skills research determined were essential, what skills interfered with or hampered instruction, and what was currently occurring in reading instruction. In the scientific method, we build our inferences from within a coherent framework of the existing knowledge base—from empirical studies and outcomes that have weathered the test of replication. Headsprout founders spent over a year reviewing the literature and determining areas of instruction, levels of instruction, and types of instruction. Using a constructional approach (after Goldiamond, 1974), we asked “What needs to be established, what repertoire needs to be built, where do we begin?” In general, we found that research on effective reading instruction focused on these critical skills (as identified by the National Institute of Child Health and Human Development, 2000):

**Critical Skills for Early Reading Repertoires**

Phonemic Awareness  
Phonics  
Fluency  
Vocabulary Development  
Text Comprehension

These skills were analyzed into their constituent elements. That is, we needed to determine where the elemental skills that comprise each of the identified critical skills fell in the context of a skill hierarch that included, strategies, principles,
Step 2. Instructional Objectives.
Based on the content analysis and identification of critical areas for reading success, we identified larger, “composite” skills that we knew our learners should be taught to do (See Figure 3).

For example we knew that “Phonics,” or the relationship between letters (graphemes) of written language and the sounds of written language (phonemes) involved the ability to rapidly identify print in the presence of sound, or to reliably produce sound in the presence of print. Table 1 provides sample “composite” skills and types of discriminations we determined student would need to do in order to read/decode at the mid 2nd grade level.
Table 1: Sample “composite” skills

<table>
<thead>
<tr>
<th>Skill</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognize that there is a one-to-one correspondence between sounds (phonemes) and print (graphemes)</td>
<td>Paired associates, multiple discriminations</td>
</tr>
<tr>
<td>Recognize that sounds make words, words make sentences, and sentences make stories</td>
<td>Limited range concepts</td>
</tr>
<tr>
<td>Identify the beginning, middle, and ending sounds of words</td>
<td>Multiple discriminations, algorithms</td>
</tr>
<tr>
<td>Segment words into sounds</td>
<td>Psychomotor chains; multiple discriminations</td>
</tr>
<tr>
<td>Use letter-to-sound correspondence to sound out new words</td>
<td>Principle applying; Psychomotor chains</td>
</tr>
<tr>
<td>Fluently blend sounds to read hundreds of new words</td>
<td>Principle applying practice</td>
</tr>
<tr>
<td>Use a repertoire of basic word families and patterns to read new words</td>
<td>Principle applying, Algorithm following, strategies</td>
</tr>
<tr>
<td>Read frequently occurring words automatically and confidently</td>
<td>Paired associate, multiple discrimination practice</td>
</tr>
<tr>
<td>Read irregularly spelled words and words containing diphthongs, special vowel spellings, and common word endings</td>
<td>Paired associate, multiple discrimination, principle applying</td>
</tr>
<tr>
<td>Self-correct reading mistakes</td>
<td>Principle applying</td>
</tr>
<tr>
<td>Independently read aloud early stories with at least 90% word recognition accuracy</td>
<td>Verbal repertoire</td>
</tr>
<tr>
<td>Recognize and use the cues of punctuation when reading - including commas, periods, question marks, and quotation marks</td>
<td>Principle applying</td>
</tr>
</tbody>
</table>

These skills involved multiple subsets of skills and behavior sequences. What does it mean to “Recognize that sounds make words, words make sentences, and sentences make stories?” We needed to determine exactly what component skills made up our composite skills. We needed to identify “pinpoints” and “learning channels.”

We examined the composite behavior, “Recognize that there is a one-to-one correspondence between sounds (phonemes) and print (graphemes)” and began to break it down into component or “tool” skills. What were the minimal behaviors needed to establish a single letter sound relationship for a learner? Some of our findings are represented in Table 2.
Table 2: Sample minimal behaviors to establish letter sound relationships.

<table>
<thead>
<tr>
<th>Pinpoint</th>
<th>Skill Hierarchy</th>
<th>Learning Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listen to pairings of auditory and visual presentations of an element.</td>
<td>Paired Associate</td>
<td>[See/(Hear), then Hear/(See)]</td>
</tr>
<tr>
<td>Make observing response to element.</td>
<td>Paired Associate, psychomotor responding</td>
<td>See•Hear/Click element</td>
</tr>
<tr>
<td>Select examples of element from example (eg) /non-example (neg) array</td>
<td>Multiple Discrimination</td>
<td>Hear element•See elements/Click element heard</td>
</tr>
</tbody>
</table>

A live, interactive sample of this sequence may be found online, at www.headsprout.com/aol/teachers.cfm.

**Step 3. Criterion Test.**

From these areas we built our criterion-based measures of learning--what we would use to test for establishing mastery, fluency, and application (See Figure 4). Our computer-based Internet delivery made the identification of the response topography easy—we were not using voice recognition, so all we had was a “mouse click.” We used our content analysis and wealth of teaching experience and determined click exit criterion for each pinpoint prior to building the instructional sequence. The typical method of building a test to determine what a student has learned after an existing instructional sequence is designed or used is antithetical to the programming process approach. Tests were built into the sequence in the form of terminal exercises. Episodic tests were designed to test the cumulative effect of an entire lesson, and finally tests were designed to determine overall student performance once finished with the entire sequence.

![Figure 4. Criterion Tests](image-url)
Step 4. Entry Repertoire:
We also had to figure out what skills our students needed to bring to the learning interaction for the instruction to be effective (See Figure 5). We had to determine what was the current relevant repertoire, or entry behavior. Several questions must be answered in assessing the learners’ current relevant repertoire and in determining the starting point for the program. These questions included:

What do the learners come in with?
Given what we are (reading/decoding) trying to achieve, what are the minimal entry behaviors?
Where should a child begin the program?
How will we know?
What pre-requisites need to be taught?

Answers are provided through the content analysis, literature reviews, further analyzing component skills, probing actual learners, and tryout and revision cycles. Our answers occasioned us to refine our original goal for our first product:

“Teach a non-reading 4- to 7-year old the reading/decoding skills typically covered by mid-first grade in 40 short lessons.”

Figure 5. Entry Repertoire
Step 5. Instructional Sequence.
We had determined what it was we thought needed teaching, and how we thought we would know if students had learned it. Then we had to figure out how to teach it. We needed an instructional sequence designed to take learners from their entry repertoire to the terminal outcome (See Figure 6). We were alternately hampered and aided by our choice to have all instruction delivered entirely over the Internet. Unlike books, the Internet allows for direct interaction with the learner's behavior. Unlike humans, no person would be available to mediate and provide contingent feedback. Continuous online delivery, though not completely foreign to us, presented us with unique challenges. Not only would we not have an adult sitting there with the child, but also we wouldn’t be able to hear a word our learners said—and yet we were teaching oral reading.

Figure 6. Instructional Sequence

The guiding principles of our design were non-linear and systemic. We analyzed not only the immediately or directly visible contingencies, but also alternative sets. We did this not only for the primary behaviors of interest, but systemically--for the matrices of behaviors that made up the repertoire.
**Nonlinear vs. Linear Analysis:**

- Analyzing not only the contingency of which the target behavior is a member (the direct or linear relations), but
- Alternative sets, or matrices, of consequential contingencies, of which the target behavior and currently available alternative patterns are members, and
- Those contingencies or relations which may potentiate the matrices, --the nonlinear relations,
- And applying this analysis to understanding the patterns of observed learner behavior, which occur as a result of the interaction of these matrices.

In a science of instruction, there is no such thing as a student not “learning,” the matrix shows the “sense” of the behavior— “errors” are treated as the rational outcome, in a behavioral economic sense, of the current program contingencies and their alternatives, including the learning history of the learner. Consider as an example of this approach as it pertains to reading instruction the situation where a learner is faced with the task of selecting a printed letter pair, such as “/ip/” that represents the spoken phonetic unit /ip/ upon hearing /ip/ from an array of other letters representing other phonetic units. This is commonly referred to as a conditional discrimination. We may observe that whenever the learner hears /ip/, the learner selects “ip,” at a percent correct rate of 90%. We may feel quite comfortable that the conditional discrimination is well established. Upon closer examination, however, we may discover that the learner has learned to reject the other stimuli used, and by exclusion pick “/p/.” Or, on some presentations choose “ip” when it is surrounded by one set of other elements, and choose away from the other elements upon different presentations. That is, hearing /ip/, seeing “ip” may only guide half the correct responses, whereas, hearing /ip/, seeing that the other sets are something other than “ip” may guide the other half. These two different contingencies may combine to produce a 90% correct rate. Further, in conditions of ambiguity the past consequences of calling something an example when it is not (a false alarm), or failing to call something an example when it is (a miss), may influence the likelihood of selecting the example stimulus as much or more than the consequences for selecting the example (Robbins, Layng, & Karp, 1995). All these contingencies, and more (see for example Blough, 1972; Sidman 1980, 1992), act together to determine any given response at any given point in time. It is the job of the learning scientists and instructional designers to consider all these factors when designing instruction, testing program segments, revising sequences, and interpreting learner data.

**Systemic vs. topical intervention**

- The instructional problem presented is not always the problem to be solved. Learner failure in a particular sequence may not be the result of the design of that particular sequence, but of the design of preceding sequences, or their order of presentation, or the amount of practice.

**Guided practice**

- Important to the effort is the programming of well-designed guided practice, which firms the skills and leads to learner fluency. Practice follows the establishment of components so that they will be readily available for recombination with other equally firm components (Johnson & Layng, 1992; 1994). Practice is often timed. Timing allows for the greatest number of response opportunities in the shortest period of time, and may have properties
in its own right that assist in the retention of skills. The timed practice and the basic instructional sequence are intertwined, one building on another.

- **Special challenges of delivering over the World Wide Web**
  - In designing for the Internet, one must always consider the equipment used by the most minimally equipped end-user. The program must run well on older equipment, on a variety of web browsers, on a variety of machines, and on a variety of bandwidths. No special equipment or configurations must be required. Instructional designers must always plan for the worst case in program delivery. Further, instructional designers must coordinate with graphic designers, scriptwriters, sound engineers, and software engineers, among others –it’s instructional design meets Hollywood!
  
  - On top of all of this, is the business side of delivery. The program must be embedded in a website, decisions about user access have to be made, and an e-commerce interface has to be built. And finally, revisions based upon learner use and technological and bandwidth innovations must be continuous. While there is no question that Internet offers tremendous possibilities and potential, the development effort needed to produce high quality instruction is enormous and very expensive.

We needed a reliable instructional sequence. What we wanted was an “instructional program.”

**Definition of an instructional program**A learner verified, reproducible sequence of instructional events designed to produce a measurable and consistent effect on the behavior of each and every targeted learner.

How would we know when we had an instructional program? We could only know by frequently and carefully analyzing performance data. Without our user test lab and continuous measurement, a nonlinear approach to curriculum design would not be possible.

**Step 6. Performance Data.**

Each element of the program had to be continuously tested with new users during development—not after. No matter how well thought-out any sequence might be, learners will show where the analysis, design or both went wrong. Very systematic try-out and revise cycles are required. Data must be used to track changes and evaluate various versions tested (See Figure 7). Fortunately, with computer-based instruction, data can be taken and used to adapt the program to the learner’s responses to help ensure mastery of the material. In addition, Headsprout's on-site user test lab allowed us to record, observe or probe for behaviors that the computer could not detect. If a student could not perform the terminal exercise for an instructional episode or sequence within an episode, the episode or sequence was revised until the student met the exit criteria. Further program extrinsic episodic testing was performed to validate that the criterion measures built into the program. Once the program sequences and episodes were learner verified, tests were administered to determine if the sequence itself met criterion. Changes were made in the program until 90% of the learners met the exit criteria. This process resulted, to date, in over 10,000 data-based program revisions.
Performance Data

The process involved, and still involves constant testing, revision and recycle, and retesting. This process is referred to as "formative evaluation" by Scriven (1974), and involves the consensus of best practices, experience, and point of view. The design is initially based on previous research, may come from a variety of disciplines, and may be based on elements implemented in the program and not on the program itself. All elements of the program are tested for effectiveness, and if the criteria are not met alternative strategies are built and tested. The processes iterates until criteria met, with performance always measured against a set of criteria. How to sequence the program steps and the relation of behavior to the sequence is explicitly identified, thereby generating new knowledge. This process continues and becomes aggregated as the "chunks" of the program units change in size (e.g. for Headsprout: a segment of a lesson, a lesson, groups of lessons, the program). The research based on individuals, and therefore can generalize to individuals (Sidman, 1960).

Our instructional objectives have been further refined, with skills added and others removed from the sequence. This process was incorporated in three distinct stages of testing (Markle, 1967): Developmental Testing (conducting in-house, in the user test lab); Validation Testing (conducted as a beta release of the program, in controlled remote locations); and Field Testing (conducted via public launch of the program).
These three phases of empirical testing (developmental, validation, and field testing) identified by Markle (1967) are briefly summarized below (and were also described by Johnson, Twyman & Hobbins, 2001):

**Developmental Testing**
The Developmental Testing Phase includes these elements:

- **Goal** - to develop a workable instructional program
- **Setting** - laboratory
- **Learners** - a small number of highly vocal students for ID to observe and interact with, one-on-one, while they learn from the program. Learners are often encouraged to talk out-loud their “reasons” for doing what they are doing.
- **Activities** - progress through initial drafts of program
  - Start only with objectives in hand (maximal learner input), all the way to a carefully sequences set of tasks (minimal learner input)
  - Tryout, revision, try-out, revision, tryout . . .
- **Data collected** - wide variety of general and precise performance and affective measures, narrative & quantitative.
- **Data-based interventions** - include debugging, improving writing, graphics, layout, motivation, and learning variables
- **Begin with leanest possible design** that might achieve the instructional objective. Let the learner's behavior show what needs to be added. Instruction is like salt, easier to put in than to take out.

**Validation Testing**
The Validation Testing Phase includes these elements:

- **Goal** - precise description of performance characteristics that occur when learning with the product, and under what conditions, in how much time.
  - product description, not product development.
  - Systematic replication across learner demographics
- **Setting** - locations as advertised
- **Learners** - as advertised
- **Activities** - progress through the program
  - teacher, boss, parent, self-supervised, as advertised.
  - ID close by.
- **Data collected** - ongoing learner performance in program, criterion tests
- **Data-based interventions** although this is a demonstration phase, data are published to indicate the extent and limit of program effectiveness, and to make further refinements in the program
  - Who does/doesn't learn?, How much is learned?, In how much time?

**Field Testing Phase**
The Field Testing Phase includes these elements:

- **Goal** monitored use of product in the field
  - Compared to other product(s)
  - With a new population or objective, not originally planned
  - In practice, this often subs as validation testing
- **Setting** widely varying, hundreds or thousands of learners
- **Activities** progress through the program, under advertised conditions
- **Data collected** learner performance on criterion- and norm-referenced tests, sometimes even more general.
- **Data-based interventions** - determine more about extent and limit of program effectiveness, used to continue to refine the program.

All three phases are employed in the development of Headsprout programs. Although very expensive and time consuming, without this commitment to user testing, it would not be possible to build truly scientifically based instruction (for more information, please see Layng, et. al., 2002). The Internet, while presenting challenges, also presents opportunities. Almost every learner response to the program can be collected and analyzed. As the number of users grows so too does the on going field-test database. When the program is revised, all learners instantly get the revision.

**Step 7. Maintaining Consequences.**
Like any instructional sequence, we needed contingencies that would maintain the individual learners’ behavior throughout the instructional sequence. Without human delivery, alternative contingencies had to be built into every step of the program. This is done by (1) ensuring the learner is successful and makes visible progress in the program, (2) by careful placement of extraneous, program extrinsic, consequences, and (3) by ensuring that the skills being learned will be useful, program intrinsic consequences (Goldiamond, 1974) (See Figure 7). Careful analysis of reinforcement schedule variables is also required. We spend a great deal of time inventing methods to combine the maintaining consequences with the optimal instructional and practice sequences.

**Maintaining Behavior Throughout the Program**

![Diagram](Diagram.png)

**Figure 8. Maintaining Behavior Throughout the Program**
Conclusion

Teaching involves coordination of a complex set of repertoires on behalf of the teacher and the student. A whole host of complex interlocking consequential contingencies must be carefully analyzed, sequenced, and arranged. Given how important the teaching of reading is to America’s future, Headsprout had no choice but to base its beginning reading program not only on the most up-to-date scientific literature on the teaching of reading, but also on the employment of state-of-the art scientific control–analysis methodology in its design.

How large a scale was the project that produced the initial Headsprout beginning reading program? It took over 25 people, 2 – 3 years, and over $4 million to produce the 40 Episode program and its underlying Generative Learning Technology. This team included learning scientists, instructional designers, prototype programmers, copy editors, creative writers, graphic artists, animators, sound engineers, system architects, system administrators, and web software engineers. All of whom had to be coordinated and synchronized as part of an iterated product development system – a nonlinear meta-system, informing and being informed by the nonlinear instructional design system.

We have demonstrated that it is possible to develop a complex cumulative repertoire (see, Johnson & Layng, 1992; 1994) over the Internet, where component behaviors are systematically established and brought together to make larger behavioral units called composite repertoires. These composites can then combine with other components or composites to make yet other composites. Whereas many approaches to computer-based instruction may appear to limit what a learner experiences, our non-linear approach frees the learner to respond in ways not typically thought possible for computer-based instruction. Indeed, we can actually engineer successful “discovery learning,” teach our young speakers to be their own listeners who often correct themselves, and reliably teach our learners to make correct articulation of sounds and words without the need for expensive, and often cumbersome voice recognition hardware and software. We have provided a highly effective and reliable teaching system that can be easily implemented and maintained. There are few limits on the ultimate scalability of the Headsprout beginning reading program. It is hoped that the Headsprout effort will serve as a model for other large-scale instructional projects whose goal is not only the design and deployment of instruction based on scientifically based principles, but instruction that is itself a product of a scientific approach to design.
References


