Pearson Digital Learning

Technical Note

TN030409

SuccessMaker® Motion: A Research Summary

Tony Thrall and Barbara Tingey

15 April 2003

.

Abstract

This paper summarizes the algorithms that govern SuccessMaker's instructional interaction with the student, and forecasted student progress reported to the teacher. The paper also cites examples of the research on which the algorithms are based.

Table of Contents

Introduction

The research underlying SuccessMaker® courseware was initiated by Professor Patrick Suppes at Stanford University during the 1960's, continued by Mario Zanotti at the Computer Curriculum Corporation (Suppes and Zanotti, 1996), and is now being extended by Pearson Digital Learning.

The purpose of the research is to emulate a human expert tutor who discerns and responds to the individual instructional needs of each student and provides essential information to the classroom teacher. Although the computer cannot fully emulate the nuanced interactions between a human tutor and the pupil, the company's ambition is to equip the classroom teacher with an instructional aid who is expert, affordable, and educationally effective.

This ambition is being realized through a series of refinements made possible by extensive analysis of student-system interaction. This paper summarizes the main components of the instructional system and cites a few examples of the supporting research.

Underlying educational models

The courseware has been designed to emulate a human tutor according to certain educational models of curriculum, student learning, individual tutoring, and the needs of classroom teaching. The models and courseware are refined as we learn more from our own analyses and those of other educational researchers.

Content Model: What should the tutor know about the subject?

How should the tutor organize and present the subject matter -- the content, knowledge structure, and processes of the subject? SuccessMaker courses are designed by curriculum developers to present the structure and content appropriate for a specified subject, level, and pedagogical approach.

Student Model: What should the tutor know about the student and learning processes?

The teacher provides the system with initial information about the student by selecting the course, level, and other courseware options appropriate for the student. Then the courseware learns more about the student through the student's responses during initial placement and subsequent instruction.

Tutor Model: How should the tutor instruct the student?

An expert human tutor continuously adjusts the content and mode of presentation based on the student's recent responses and the tutor's instructional goals and prior knowledge of the student. Similarly, the courseware presents the student with objectives at selected levels from a mix of curriculum strands in a selected mode, be that a question, a tutorial, brief feedback, or other forms of instruction. The tutor model thus combines the content model and the student model, guided by the principle of engaging and pleasantly challenging the student as the student gains knowledge of the subject.

Teacher Model: How should the tutor collaborate with the classroom teacher?

To maximize student learning the tutor and classroom teacher need to work together. Therefore the tutor, as emulated by the courseware, should adapt to the teacher's goals for the class and for the individual student, which is accomplished by the teacher's adjustment of strategies for the student within a course or across courses. In addition, the tutor should share with the teacher information about the academic progress of the student, which is accomplished through the courseware reporting system. Based on this information, the teacher can decide when to give the student more freedom or more structure; more direct instruction or more explorations; more focus or more variety. Alternatively, the teacher can let the program adjust to the student and make the decisions about what to do next.

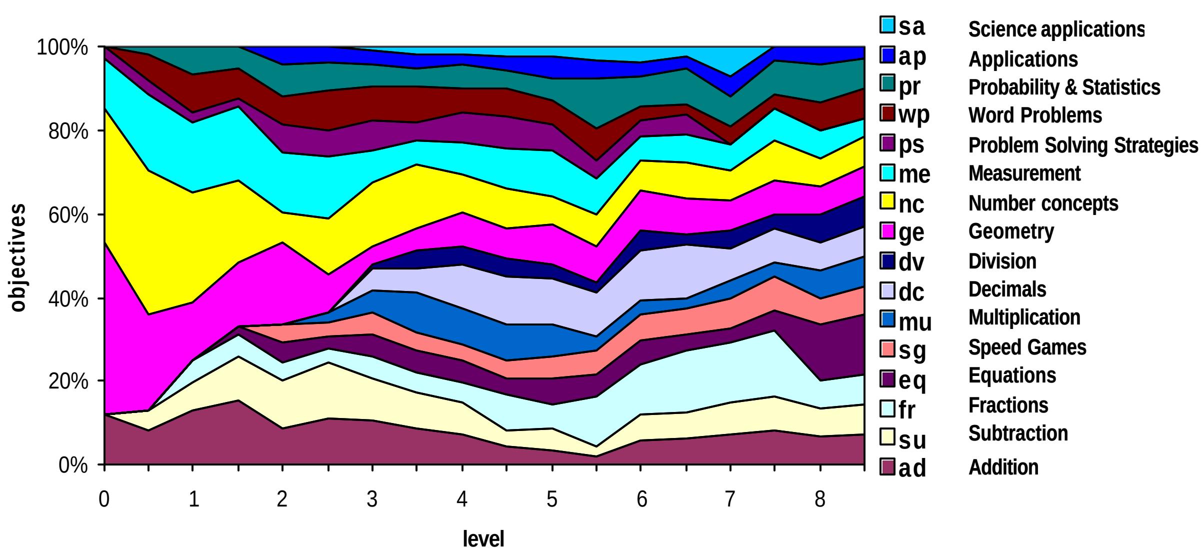
Content model: curriculum structures

The content model is embodied in the curriculum structure of each course. The broadest features of this structure are the scope and sequence of topics within a course, and the finest structural features consist of highly specified learning objectives, along with the dependencies among these objectives.

Gross structure: levels and strands

The content of a SuccessMaker foundations course is organized by concept area ("strand"). Each strand contains material spanning several years of school instruction. A strand is composed of learning objectives, with each objective identified by course level, corresponding to the level of difficulty of the material.

Figure 1: Learning objectives per strand at each level -- Math Concepts and Skills



This architecture is exemplified by the Math Concepts and Skills (MCS) course. Figure 1 shows the percent of objectives contributed by each strand for course levels ranging from 0 to 9, corresponding to a school math curriculum spanning grades K-8. The course consists of 8 computational strands and 8 application strands.

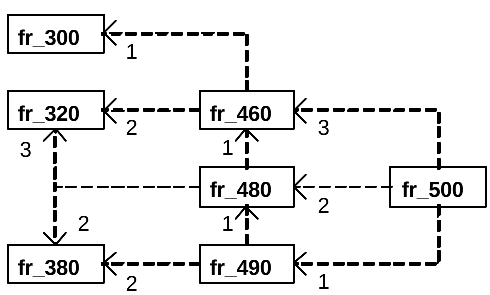
In Figure 1, starting at the bottom the following eight computational strands are introduced at successively higher levels: addition, subtraction, fractions, equations, speed games, multiplication, decimals, and division. Similarly, the eight application strands are introduced in the following order: geometry, number concepts, measurement, problem-solving strategies, word problems, probability and statistics, general applications (such as interpreting the numeric information displayed in a chart), and science applications.

Fine structure: objectives and prerequisites

Strands or concept areas are the largest curriculum structures in a SuccessMaker course, and objectives are at the finest level of granularity. The Math Concepts and Skills course, as a prime example, contains over 1600 highly specified learning objectives.

As an illustration, Figure 2 displays 7 objectives within the fractions strand. The arrows in the figure denote a prerequisite relationship. For example objective fr_500 (change a mixed number to a fraction) requires three prior objectives: (1) fr_490 (change a fraction to a mixed number); (2) fr_480 (change a whole number to a fraction); and (3) fr_460 (add fractions with like denominators). The objectives are labeled by strand and level. For example, fr_500 is a fractions objective at level 5.00. The ordering of prerequisites (indicated by the number in the figure next to each arrowhead) denotes the sequence in which prerequisite knowledge is to be checked (as mentioned below and as described more fully in Suppes and Zanotti, 1996).

Figure 2: Cascade of prerequisites for fr_500 (change a mixed number to a fraction).



Instructional exercises versus test items

For each detailed learning objective the courseware generates a set of equivalent exercises that differ in certain randomly selected variables and graphic presentations. This variation of exercises within an objective resembles the variation of related items across equivalent forms of a test, a resemblance that can lead to the misidentification of courseware exercises as test items.

Moreover, the courseware, in its emulation of a human expert tutor, tacitly assesses student understanding during initial placement of the student in the curriculum and subsequent evaluation of the student's mastery of objectives. Nevertheless, tutoring and testing can be clearly distinguished by their primary goals of instruction and assessment, respectively. The tutor errs on the side of heading off confusion and promoting understanding. For example, courseware exercises are designed to detect and address a student's errors as quickly as possible, whereas test items are designed to measure the student's current knowledge as accurately and reliably as possible. These distinct instructional and assessment goals lead to distinct lines of research, development, and product evaluation.

Student model: learning and remembering

The courseware represents the student's learning of the curriculum by formulating the notion of student understanding and interpreting the pattern of student responses to exercises. Specifically, the student's understanding of a learning objective is formulated as a probability that the student will correctly respond to the next exercise pertaining to the objective. This probability is updated based on the sequence of student responses according to the following assumptions:

(1) The student's most recent response, if correct, indicates an increased probability of success on the next exercise;

(2) The student's more recent responses are better indicators of the student's current understanding than are earlier responses.

Suppes and Zanotti (1996) discuss various probability models with respect to these assumptions. The specific models on which the courseware is based are adjusted as the courseware changes and as student performance on the courseware is analyzed.

The courseware also addresses the possibility that a student may initially master a learning objective and then forget it or fail to apply it. For example, if a student has repeated difficulties with a new learning objective, the courseware responds with various strategies, including a review of prerequisite learning objectives. Additionally, in the Math Concepts and Skills course, previously mastered computational skills are reviewed systematically to ensure the student's continued computational fluency. (See the discussion of instructional strategies below.)

Tutor model: instructional algorithms

The curriculum structure and content can be regarded as the emulated tutor's subject-area knowledge. The learning models and data constitute the emulated tutor's knowledge of the typical student's mode and rate of progression through the entire curriculum, along with the individual student's performance to date. An effective tutor must use this information in a variety of ways to instruct the student, for example:

- Decide where to begin instruction;
- Determine when the student demonstrates learning;
- Emphasize content appropriate to the student's needs;
- Clarify points when difficulties arise;
- Remind the student of knowledge previously attained;
- Check the student's recollection of previously mastered material.

The following summarizes the courseware algorithms that accomplish these instructional tasks.

Decide where to begin instruction: Initial Placement Motion

SuccessMaker's Initial Placement Motion (IPM) is designed to find the right level for learning, between what the student knows and doesn't know. This placement mode of operation differs from the instructional mode not in the content used but in the variation of learning tasks presented to the student.

That is, during normal instruction the system minimizes the variation in levels among the student's active strands by presenting relatively more material from the student's lowest strands. In addition, continuity of instruction is maintained by permitting only gradual changes in the level of the material (although a student may rapidly demonstrate mastery at each level).

During placement, however, these restrictions are relaxed considerably, so that the appropriate course level may be found as quickly as possible.

Assess the student's initial mastery of an objective

The courseware estimates and updates the student's probability of correct response to a learning objective based on the model of student learning previously described. The courseware records student mastery of the objective once the probability reaches a predefined threshold.

Emphasize content appropriate to the student's needs

SuccessMaker courses are currently designed to keep the student's lowest strand level within a prescribed range of the average of all the student's active strand levels. This is accomplished by selecting the student's lower strands with relatively greater frequency than the higher strands. Suppes and Zanotti (1996) describe the details of this algorithm.

An advantage of this strategy is that the student gains understanding of the course curriculum as a whole. In addition, the student's average level in the course fairly represents the student's position in each of the active strands.

Adapt instructional strategies to the student's pattern of response

As previously mentioned, the courseware assesses the student's mastery of a learning objective by calculating the probability that the student will answer the next exercise correctly. More generally, the calculated probability and the number of exercises attempted by the student determine the courseware's decisions to change the learning objective, strand, or mode of presentation. Table 1 summarizes these decision rules.

Table 1: Courseware response to student performance.

Student Performance		**Courseware Response**	**Description**
Excellent		mastery	Record mastery; select and present next learning objective.
Good		mixed presentation	Present another exercise from the same learning objective; then select a new strand.
Moderate		sequential presentation	Present a sequence of exercises from the same learning objective.
Low	1	tutorial	Present any tutorials associated with the objective.
	2	prerequisite review	Present prerequisite objectives.
	3	delayed presentation	Mark the objective for later presentation.
	4	move on	Record the objective as not mastered, but move the student to the next objective.

Check the student's recollection of previously mastered content

In the Math Concepts and Skills course, the system checks the student's maintenance of computational skills, and reviews these learning objectives as necessary. The program calculates and reports a Computational Retention Index (CRI), which is the percent of previously mastered computational skills that remain within the student's mastery.

Teacher model: diagnosing and forecasting student performance

As requested by the teacher, the SuccessMaker reports show the status and forecasted progress of each student. The courseware reports the learning objectives in which the student is having the greatest difficulty, thereby summarizing the current student-courseware interaction and facilitating instructional support outside the courseware system.

In addition the courseware reports the student's rate of progress through the curriculum, and estimates the additional time the student will need to spend to achieve specified gains within the courseware. The estimation is based on time-gain analyses of students previously enrolled in the curriculum. The initial estimate, provided in a Prescriptive Scheduling report, is based on the median of previous students' trajectories through the course. Subsequent estimates, presented as an Individualized Prescriptive Strategy (IPS), modify the median trajectory based on the individual student's pattern of progress.

Supporting Research

For decades, continuing to the present, the company has made a practice of analyzing data from student-courseware interactions in order to gain insight into student learning, refine the various models described here, and improve the courseware. The following examples from company studies illustrate the types of analysis conducted.

Probability models of learning

Probability models are used to assess the student's mastery of learning objectives, and thus play a central role in the courseware. Suppes and Zanotti (1996) describe a three-parameter model and show how the model determines the chance that a student will have any prescribed sequence of correct and incorrect answers to exercises within a learning objective. The authors go on to compare modeled probabilities to the respective percents of students with each possible pattern of responses to a sequence of four exercises. The consequent chi-square tests of the model confirm the model's close agreement with actual student percentages.

The parameters of the Suppes-Zanotti model signify, respectively, the difficulty of the learning objective (q), the relative importance of the student's most recent response (ω), and the overall rate ($1-\alpha$) at which the student learns even when responding incorrectly. The continuing refinements of these probability models remain amenable to interpretation, which is essential for sharing knowledge among teachers, curriculum developers, and educational researchers.

Efficiency of instructional strategies

Another continuing line of development is the crafting of exercises and instructional strategies within a learning objective. If done well, the student should be able to master a learning objective quickly.

Thus the efficiency of the courseware can be measured from the number of exercises the average student must attempt to achieve mastery, or from the percentage of students who respond incorrectly at each successive attempt. The percentage may start off higher for the more difficult learning objectives, but in any case should drop off rapidly with successive attempts.

Suppes and Zanotti (1996) provide several examples showing an approximately geometric decline in the percentage, with the multiplicative constant of decline (α) on the order of 0.8 to 0.9. A continuing research goal is to find ways to make this coefficient as small as possible.

Student trajectories through the course

No matter how rapidly students proceed through a course on average, and no matter how well the courseware adapts to the individual student, students will require differing amounts of time to achieve a specified degree of progress, i.e., to raise their respective course levels by a specified amount.

Accurate estimation of the amount of time each individual student requires is important for allocating, monitoring, and revising courseware time among students. Increasingly, teachers and administrative staff are using the courseware's forecasting capabilities to manage student achievement, not only within the courseware but also on state assessments, to which courseware levels are statistically correlated.

Suppes and Zanotti (1996) show how widely individual trajectories may vary, and how students currently at the same level in a course may have quite distinct forecasts. The authors model individual student trajectories as power curves. For each student in their study, they calculate the forecasted and actual gain at successive time points, from which they calculate the mean absolute deviation, a measure of the degree to which the model approximates the data. The resulting distribution of mean absolute deviations across all students in the study shows the model to approximate the data quite closely.

Further analyses have extended these power curve models and led to individual estimates based on percentiles of the distribution of trajectories. For example, a student whose current trajectory corresponds to the trajectory at the 25th percentile of the distribution is forecasted to progress according to the 25th percentile.

Summary

SuccessMaker courseware is based on research begun in the 1960's at Stanford University and continuing through the present. The goal of the research is to emulate a human expert tutor who supports the main instruction of the classroom teacher. The emulated tutor's expertise resides in its understanding of the curriculum to be taught, the needs of the student, the strategies of instruction, and the requirements of the classroom teacher.

The research culminates in a set of courseware features. In particular, the sequence of presentation of content to the student (the "courseware motion") is guided by a system of research-based algorithms. In the SuccessMaker foundations courses, the system consists of the following components (all of which are present in the Math Concepts and Skills course).

- Initial Placement Motion (IPM) finds the student's appropriate level in the course, a level that is appropriate for learning, neither too easy nor too difficult for the individual.
- Mastery decisions are based on the probability of the student answering the next exercise correctly, not merely on the student's current percentage of correct answers. The courseware thereby responds more quickly to student understanding, resulting in a more efficient use of the student's time.
- Dynamic sequencing of content adjusts to the individual student. When the student experiences repeated difficulties with new material, the material is set aside ("delayed") for subsequent presentation. Again, the goal is to challenge the student without frustrating him, and thereby to keep him engaged in the courseware.
- The proportion of instruction across concept areas is adjusted for the individual so that weaker areas receive more emphasis, thereby reducing the gap between the student's areas of relative weakness and strength.
- Tutorial intervention guides individual student learning. When the student encounters difficulties the system employs various instructional strategies, including sequential practice within the area of difficulty, presentation of brief tutorials, and/or review of prerequisite material.
- By periodically checking the student's recollection of previously mastered material, the system assures the student's firm basis for further learning.
- The time a student requires to achieve specified gains is estimated and reported to the teacher. The estimate is initially based on the data from past users of the courseware. Then, as the system analyzes the individual student's rate of progress, it adjusts that estimate.

The courseware thus constitutes a learning system that adapts to the individual student. The curriculum structures, motion algorithms, and reporting functions that make this possible are designed to engage the student and assist the teacher. These product features are the consequence of continuous model refinement based on a sustained program of research.

References

Suppes, P. and Zanotti, M. (1996). Foundations of Probability: Selected Papers, 1974 - 1995. New York: Cambridge University Press.