Teaching Tip

An Approach to Reducing Cognitive Load in the Teaching of Introductory Database Concepts

John M. Bunch
The SAJES Group, Inc.
17345 Emerald Chase Dr.
Tampa, Florida 33647 USA
jmbunch@sajes.com

ABSTRACT

This paper presents a goal-based scenario approach to teaching introductory database concepts to undergraduates using two different scaffolding methods. One method, termed *worked-out examples*, attempts to reduce extraneous cognitive load by requiring students to complete increasingly complex missing parts of worked out examples. The other method, termed *progressive practice*, attempts to reduce intrinsic cognitive load by requiring students to complete increasingly more complex scenarios, beginning with a simple example and culminating in an alternative form of the target scenario. Classroom results with these methods have been positive, and are discussed.

Keywords: Problem-based Learning, Constructivism, Scaffolding, Cognitive Load, Database Concepts

1. INTRODUCTION

As noted by Merrill (2002) in his synthesis of the past several years of instructional design theories and models, the engagement of learners in real-world problems promotes learning. Merrill defines a real-world *problem* as one in which the learning activity represents a whole task, and the task is representative of problems the learner would encounter in the real world. Further, Merrill contrasts *problem-centered* instruction with the more common *topic-centered* instruction, in which decontextualized skills are taught first, with the assumption that real-world problems can be solved once the learner integrates these prerequisite, independently-taught skills.

Connolly & Begg (2006) argue that problem-based instruction can be particularly helpful in teaching the abstract concepts inherent in database design. Connolly & Begg (2006) describe the difficulties in effectively teaching design concepts in the traditional teacher-as-lecturer format, and suggest that the learning-by-doing model of problem-centered instruction is needed to effectively impart these skills.

A problem-centered instructional method that seems well-suited to using real-world problems in the teaching of both the process and factual knowledge inherent in information systems education including database concepts, is the *goal-based scenarios* (GBS) method of Schank, Berman, and MacPherson (1999). in which students work toward a target goal by applying both the process and factual knowledge that form the basis of the lesson. This paper summarizes an approach I have used to teach fundamental concepts of normalized database design to an audience of novice, undergraduate learners using goal-based scenarios.

1.1 Approaches to Reducing Cognitive Load in Problem-Based Instruction

As noted by van Merrienboer, Kirshner, and Kester (2003), a potential problem with problem-based instructional approaches is that the learner becomes overwhelmed by task complexity, causing a level of cognitive load that interferes with learning. As discussed by Tuovinen & Sweller (1999), high cognitive load occurs when the many elements of instructional material compete at once for cognitive resources – primarily those of working memory. Paas, Renkl, & Sweller (2003) note that *intrinsic cognitive load* refers to the interactivity of the elements of the material being learned, while *extraneous cognitive load* refers to demands on working memory brought about by the instructional materials themselves (for example, the requirement to search for a problem solution). All forms of cognitive load are additive, and for learning to occur they must not exceed the available cognitive resource capacity of the learner.

An instructional approach useful for creating appropriate levels of cognitive load is *scaffolding* – presenting performance supports as needed to achieve a goal, then fading them away as the learner is able to achieve the goal on his or her own (van Merrienboer, et al., 2003). Van Merrienboer, et al. (2003) describe two approaches to scaffolding whole tasks for the reduction of intrinsic and extraneous cognitive load. In the first approach (which for purposes of this paper I will call *progressive practice*), intrinsic cognitive load is reduced by having learners first begin working on simple versions of the whole task, and then progress through increasingly more complex versions until the target task is accomplished. In the second approach (termed *worked-out examples, i.e Tuovinen & Sweller, 1999),* extraneous cognitive load is reduced by having learners complete small parts of worked-out examples, then progressively complete larger parts of additional worked-out examples until the target task is accomplished.

Many authors have noted positive results in the use of worked-out examples for the acquisition of complex cognitive skills across a variety of domains (i.e. Hilbert, Renkl, Schworm, Kessler, & Reiss, 2008; Yaman, Nerdel, & Bayrhuber, 2008; see Pass & van Gog, 2006, for a review), and note that the modularization of problem solutions into smaller solution elements (inherent in both scaffolding approaches presented here) itself works to reduce cognitive load (Gerjets, Scheiter, and Catrambone, 2004).

While there is support for the use of scaffolding in information systems education in general (i.e. McFarland, 2003; Linder, Abbott, & Fromberger, 2006; Garner, 2007), there is no clear indication of which of these scaffolding techniques would create the most efficient and efficacious instructional delivery method for teaching basic database concepts to an audience of novice learners. Tuovinen & Sweller (1999), in a comparison of worked-example versus discovery learning approaches to teaching basic database concepts using FileMaker Pro, found that novice learners benefited greatly from the worked-out example approach. Discovery learning, however, lacks the goal-based structure implicit in both examples and progressive practice.

In the remainder of this paper I present both progressive practice and worked-out examples versions of a series of scaffolded, goal-based scenarios developed by this author to teach basic database concepts to undergraduates enrolled in an "Introduction to Computers and Technology" class, part of a college transfer curriculum at a large urban community college in the US. In addition, I present a comparison of course performance and student preferences between course sections using the progressive practice vs. worked-out examples scenarios.

2. INSTRUCTIONAL MATERIALS FOR A SCAFFOLDED, GOAL-BASED SCENARIO APPROACH TO TEACHING BASIC DATABASE CONCEPTS

2.1 Learning Goals

Following Schank, Berman, and MacPherson (1999), learning goals for the database unit were divided into two categories – the process knowledge required to successfully practice the desired skills and the content knowledge necessary to achieve the goal (see Table 1). The target scenario is shown in Table 2. This scenario served as the final goal-based scenario to be completed, and was presented as an identical task in both the progressive practice and worked-out examples series.

2. 2 Scenario Formats and Supplemental Materials

Two different series of scenarios were developed to achieve the learning goals listed in Table 1. The first used the worked-out examples scaffolding technique, and the second used the progressive practice scaffolding technique. There are five scenarios in each sequence, and each is designed to be a stepping stone to the next scenario in sequence, leading ultimately to the target scenario. Table 1 also lists the incremental skills required to complete each scenario.

While the scenario sequences presented here are intended to be the centerpiece of the database instructional unit, additional instruction in the content and procedural knowledge necessary to complete the scenarios is also needed. However there are few constraints on the delivery format of this material, and it is conceivable that such material could be delivered, for example, through traditional face-to-face lectures and demonstrations, textbook readings, self-paced and Web-based tutorials, or any other delivery format that fits the situational delivery needs. In addition, while I use MS Access as the database platform in which the scenarios are implemented, any contemporary RDBMS platform would work equally well, and may be preferable depending on continuing curriculum or vocational needs.

What is critical is that students be given timely and specific feedback on the accuracy of their scenario solutions before they progress to the next scenario in the sequence. The goal is that a student will master the material contained within each scenario before progressing to the next.

2.2.1 Worked Out Examples Sequence - The worked-out examples sequence features scenarios of the same level of complexity and requiring the same skills and content knowledge as the criterion scenario. In other words, each scenario is designed to be an alternate version of the criterion scenario. The first scenario includes a nearly completely worked-out solution, with each successive scenario containing an increasingly less complete solution, thus requiring increasingly more complex work from the student. By the presentation of the criterion scenario all worked-out example scaffolding has been withdrawn.

Table 3 lists the first worked-out example scenario. As the first scenario in the sequence, it provides a nearly complete solution, requiring only that the student complete one table definition by identifying appropriate fields requiring integer, currency, and variable length string data types. In addition to the knowledge required by the first scenario, the second requires the implementation of a unique row identifier and additional data types. The third requires the establishment of referential relationships, the fourth requires single-table SELECT queries, and the fifth requires multi-table SELECT queries.

2.2.2 Progressive Practice Sequence - As seen in Table 4, the progressive practice sequence begins with the simplest scenario, but one that attempts to tap exactly the same content and process knowledge as the first scenario in the worked-out example sequence. This is the case for each successive scenario in the sequence, although it is important to remember that the reason for this results from a planned, gradual withdrawal of scaffolding for each sequence rather than an explicit attempt to make the two sequences match.

2.3 Implementation

At the beginning of the unit, students are assigned textbook readings (Chapter 9 of Shelly, Cashman, and Vermaat, 2008) – a one-chapter topical summary of the conceptual knowledge covered by the learning objectives. In addition, students are assigned the self-paced lab exercises on MS Access from MS Access Chapters 1, 2, and 3 of Shelly, Cashman, & Vermaat, M. E. (2007), which presents all of the procedural skills required to complete the criterion scenario. Finally, bi-weekly 45 minute lectures on the required content knowledge are delivered to each section.

Individual course sections are given either the progressive practice version or the worked out example versions of the scenarios, and are required to complete and submit them at a rate of about two per week. These scenarios are evaluated by the instructor and returned to the students with written feedback within 48 hours via email. As other authors (i.e. Connollly & Begg, 2006) have noted, the role of the instructor in this context is a radically different proposition than that found in the traditional teacher-as-lecturer model. Here the instructor is a facilitator, helping coach students toward a satisfactory solution to each successive scenario. Each item cannot simply be assigned a grade with mistakes red-lined – the instructor-facilitator must be prepared to assess solutions and coach students when necessary.

I use the amount of coaching needed as an indication of the appropriateness of a given *graininess* of a scaffold for a

Process knowledge	**Content knowledge**
Create an MS Access database file	Definitions and relationships between tables, fields, and records
Create an MS Access table	Field data types
Select appropriate field data type	Normalized database design
Select field length	Null values
Set field properties including allow nulls, default values, and primary key	Default values
Create table relationships	Referential integrity
Create and save queries using graphical query tool	Primary keys and foreign keys
Execute saved queries	Database queries

Scenario	*Incremental competencies*
1	Analyze business requirements and define table, create and set field length for text data type field, create integer field, create currency field, set allow null parameter for field.
2	Create primary key with auto-incrementing default value, create date/time field, create foreign key field
3	Define and implement referential integrity for database (relationships in MS Access)
4	Create query to retrieve single field constrained by WHERE clause (set query "criteria" in MS Access Queries:Design View)
5	Create query to retrieve multiple fields constrained by WHERE clause (set query "criteria" in MS Access Queries:Design View), including computed value field (using MS Access DateDiff function).

Table 1: Process and content knowledge learning goals, and incremental competencies required for each scenario in sequence (scaffolding sequence framework)

Tampa Bay Air

Tampa Bay Air is an airline company that flies routes between Tampa and several destinations in the Caribbean. It owns and operates several different airplanes and employs many pilots. Each destination has an airport name, a country name, and a distance from Tampa. Each airplane has a seating capacity, a range (in miles) it can fly on a tank of fuel, and an average airspeed (in miles per hour). Each pilot has a first name, a last name, a phone number, and an address with street, city, state, country, and postal code.

You have been asked to create an Access database for Tampa Bay Air. Tampa Bay Air wants to be able to store information about airplanes, destinations, and pilots.

Tampa Bay Air wants to maintain a list of flights made. This list should include the airplane, the destination it flew to, the pilot who flew the plane, and the date of the flight.

Finally, Tampa Bay Air wants to be able to produce a report of all flights made to a given destination. This report should include the seating capacity of the airplane making the flight, the name of the pilot who flew the plane, and the estimated flight time based on the average airspeed of the plane and the distance of the destination from Tampa.

You need to create two files:

1. Create a plan for the database in MS Word. This document should display the conceptual schema for the database, including all tables. It should also indicate the primary and foreign keys.
2. Implement the conceptual schema in MS Access. Create the tables using the appropriate data types, primary keys, and relationships. In addition, add sample data to the tables. Create a query to produce the needed report.

Table 2: Target Scenario

Downtown Flowers and Gifts

Downtown Flowers and Gifts is a flower shop that offers a variety of off-the-shelf flower arrangements, especially for holiday occasions. Downtown Flowers and Gifts buys specific flowers at wholesale to put in the arrangements. Flowers have a common name, a scientific or Latin name, a color, a wholesale quantity, and a wholesale cost for the quantity. Each flower arrangement has a name, a description, and the name of a graphic image file containing a picture of the arrangement. Each holiday has a name, a date, and a description.

Downtown Flowers and Gifts has asked you to create a database for them. In addition to storing information about flowers, arrangements, and holidays, Downtown Flowers and Gifts also wants to be able to store a list of flowers in each arrangement. This list should contain the name of the arrangement, the common name of each flower in the arrangement, the quantity of that in the arrangement, the color of that flower, and the wholesale unit cost of that flower. The wholesale unit cost is the wholesale cost divided by the wholesale quantity. Also, Downtown Flowers and Gifts wants to be able to retrieve a list of holiday specials. Holiday specials include an arrangement name, an associated holiday, an arrangement description, and a holiday special retail price. The holiday special retail price is the sum of the wholesale cost of each flower in the arrangement times the quantity of that flower in the arrangement, plus a 25% markup.

Table Name: Flowers

Field Name	Data type	Field length	Allow Null?	Default value
FID	number	integer	no	increment by 1

Table Name: Arrangements

Field Name	Data type	Field length	Allow Null?	Default value
AID	autonumber	integer	no	increment by 1
Name	text	25	no	
Description	memo	memo	no	
PictureFile	text	50	no	

Table Name: Holidays

Field Name	Data type	Field length	Allow Null?	Default value
HID	autonumber	integer	no	increment by 1
Name	text	25	no	
Description	memo	memo	no	
HolidayDate	DateTime	DateTime	no	

Table Name: FlowersInArrangements				
Field Name	**Data type**	**Field length**	**Allow Null?**	**Default value**
ItemID	autonumber	integer	no	increment by 1
AID	text	25	no	
FID	text	25	no	
FlowerQuantity	number	small integer	no	
Flower Color	text	15	no	
FlowerUnitCost	money	money	no	

Query Name: HolidaySpecials			
Field Name	**Data type**	**Field length**	**Value**
ArrangementName	text	25	arrangements.Name
Holiday	text	25	holidays.name
Description	number	small integer	arrangement.description
RetailPrice	number	integer	sum(ArrangmentName.FlowerUnitCost X ArrangementName.FlowerQuantity) X 1.25

Relationships			
Primary Keys		**Foreign Keys**	
Table	**Field**	**Table**	**Field**
Flowers	FID	FlowersInArrangements	FID
Arrangements	AID	FlowersInArrangements	AID
Holidays	HID		
FlowersInArrangements	ItemID		

Table 3: Worked-Out Example Scenario 1

Downtown Flowers and Gifts

Downtown Flowers and Gifts is a flower shop that offers a variety of off-the-shelf flower arrangements, especially for holiday occasions. Downtown Flowers and Gifts buys specific flowers at wholesale to put in the arrangements. Flowers have a common name, a scientific or Latin name, a color, a wholesale quantity, and a wholesale cost for the quantity.

Downtown Flowers and Gifts has asked you to create a database in which they can store information about the flowers they sell.

Create a text document in which you specify a sample schema for the database. For tables, list a table name, then list the field names, the data type for each field, the length of each field, whether to allow null values in the field, and a default value (if any) for the field.

Once you've created your schema, create an MS Access database in which you implement the schema.

Table 4: Progressive Practice Scenario 1

specific audience of learners. If a criterion problem is broken down into a finely-grained scaffold, with many successive steps, and all students successfully complete each one with no coaching needed, then I will re-work the scenarios in order to increase the grain by reducing the total number and making each scenario tap slightly more content and process knowledge. Conversely, if too many students require coaching at each step then I will decrease the grain by adding more scenarios and simplifying each one. For the materials presented here the number of successive scenarios was determined simply by my previous experience teaching this material to this particular audience, and making a judgment based on that.

Students receiving the worked-out examples scenarios tend to ask more questions at the beginning of the sequence then taper off. These students tend to spend a good bit of time in the beginning digesting the problem boundaries, which has the effect of lowering extraneous cognitive load on the remaining scenarios when the more complex concepts are brought to bear. I've noticed the opposite pattern for those given the progressive practice sequence – many students find the first scenario nearly trivial, but each successive scenario brings with it an ever larger problem space.

While each of these practice scenarios had a specific submission deadline in order to keep students tracking through the unit at an appropriate pace, a missed deadline had only a minor impact on a student's overall grade. The target scenario was presented as a graded knowledge assessment to all students at the conclusion of the database unit, and represented approximately one fifth of the student's final semester grade. A score from one to 100 was assigned based on the number of scenario solution features correctly implemented in both the written scenario and implemented as an MS Access database.

3. INITIAL FEEDBACK AND CONCLUSIONS

3.1 Initial Use of the Materials and Assessment

During one recent semester a formative assessment was made by giving one course section the progressive practice scenarios, a second section the worked-out example scenarios, while a third were given both and asked to choose and complete one from either group to submit by each deadline. Fifty-one students in total participated, and each section was taught by the same instructor. In addition to the graded assessment of the target scenario, a ten-item attitude assessment with five-point Likert-type responses was administered to all students to assess student attitudes and opinions concerning the methods of instruction used.

3.2 Instructional Effectiveness and Student Reactions

The criterion scenario was graded on a 100-point scoring rubric. For all students the overall mean was 90.9 (SD=8.48), indicating a generally satisfactory mastery of the material. One-way ANOVA on scores indicated no significant differences between groups ($F < 1$). As can be seen in Table 6, subjective impressions of the usefulness of the scenarios were generally quite positive and similar between groups, indicating that neither scaffolding approach appears to offer a significant advantage over the other.

The only group differences found on any measure were on two of the opinion rating items. ANOVA on the item *The material we covered about databases was very difficult,* produced $F(1, 48) = 4.96$, $p = .009$, and post hoc tests using the Student-Newman-Keuls test ($p < .05$) for homogeneous subsets indicated that students given the option of choosing scenario versions rated the material as more difficult than those given either worked-out examples or progressive practice versions. Similarly, a significant difference was found for *I completed all of the scenario assignments by the due dates,* $F(1,48) = 3.35$, $p = .043$, on which the Student-Newman-Keuls test ($p < .05$) indicated that those choosing a scenario version were more likely to give this item a lower rating that those simply assigned a version. Perhaps those students given the additional task of choosing a scenario spent time evaluating solutions for each method prior to choosing one to submit. This may have had an additive effect on overall cognitive load, making the material seem more difficult and the learning tasks more time consuming thus leading more students in this group to miss submission deadlines.

Students in the mixed-approach method were split regarding a preference for one scaffolding type over another. When asked (following completion of the database unit) which type of scenario they found most helpful, of the 12 students providing a response, three mentioned a preference for worked-out examples and four a preference for progressive practice. The remaining five students indicated that they had incorporated the use of both, with no preference for one over the other.

Aside from these few differences, comments tended to be universally positive, with no discernible qualitative differences between groups. When asked to provide comments on the use of the scenarios (i.e. *Were the scenarios helpful? Not helpful? Difficult? Easy?*), students indicated that the scenarios successfully established a target goal for learning, and gave structure to the task of acquiring the required content and process knowledge. Below are a collection of student responses representative of the issues mentioned.

- *Databases are very difficult. The scenarios were helpful, in being able to look back and see how the database is supposed to be constructed.*
- *I feel as though the information I learned about databases was presented in the right way(using scenarios) therefore it made it very easy to comprehend.*
- *The Scenarios being "real" life issues were very helpful in putting into practice the material covered. I prefer to learn with examples.*
- *I thought the scenarios were very helpful. Without the scenarios, I wouldn't have known what I was doing. The scenarios made it realistic which to me helped make it easier.*
- *Using the scenarios was really useful because it grounded the concept of databases. Databases are already tricky, but more because they're just abstract.*

3.3 Conclusions

While these findings are focused on the use of goal-based scenarios in teaching basic database concepts, it should be a straightforward task to successfully implement this technique across a wide range of information systems topics with similar results. As Connolly & Begg (2006) discuss, problem-centered instruction provides an effective approach to teaching the *design* process inherent in the database professional's task of translating business problems into a relational database. Garner (2007) reports success with a technology-based tool that uses scaffolding to support students learning basic programming concepts in an introductory programming course. Linder, et al. (2006) make a similar argument to that of Connolly & Begg (2006) but in the arena of software design, and discuss their use of scaffolding to successfully transition students from the basic concepts encountered in introductory computer science courses to the more complex projects students encounter in upper-level courses and the working world.

Ultimately, students want a practical purpose and goal to help organize and focus what they are being asked to learn. Goal-based scenarios provide an instructional effective method of providing this.

4. REFERENCES

Connolly, T., & Begg, C. (2006). A constructivist-based approach to teaching database analysis and design. *Journal of Information Systems Education, 17 (1)*, 43-53.

Garner, S. (2007). An exploration of how a technology-facilitated part-complete solution method supports the learning of computer programming. *Issues in Informing Science and Information Technology, 4*, 491-501.

Gerjets, P., Scheiter, K., and Catrambone, R. (2004). Designing instructional examples to reduce intrinsic cognitive load: Molar versus modular presentation of solution procedures. *Instructional Science, 32*, 33-58.

Hilbert, T. S., Renkle, A., Schworm, S. Kessler, S., and Ress, K. (2008). Learning to teach with worked-out examples: a computer-based learning environment for teachers. *Journal of Computer Assisted Learning, 24*, 316-332.

Linder, S., Abbott, D., and Fromberger, M. (2006). An instructional scaffolding approach to teaching software design. *Journal of Computing Sciences in Colleges, 21 (6),* 238-250.

McFarland, R. D. (2003). Teaching students to learn in the computer science and information systems curriculum: creating a distinction between content and methods. *Journal of Computing Sciences in Colleges, 19 (1)***,** 235-245**.**

Merrill, M. D. (2002). First principles of instruction. *Educational Technology Research and Development, 50 (3),* 43-59.

Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: recent developments. *Educational Psychologist, 38 (1)*, 1-4.

Paas, F. & van Gog, T. (2006). Optimising worked example instruction: Different ways to increase germane cognitive load. *Learning and Instruction, 16*, 87-91.

Schank, R.C., Berman, T.R. & Macperson, K.A. (1999). Learning by doing. In C.M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory (Vol. II)* (pp. 161–181). Mahwah, NJ: Lawrence Erlbaum Associates.

Shelly, G.B, Cashman, T. J., & Vermaat, M. E. (2007). <u>Microsoft Office 2007: Introductory Concepts and Techniques, Windows XP Edition</u>. Boston, Thomson Course Technology.

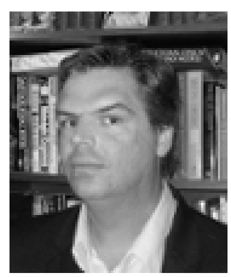
Shelly, G. B., Cashman, T. J., & Vermaat, M. E. (2008). *Discovering Computers: Fundamentals*. Boston, MA: Thomson Course Technology.

Tuovinen, J. E., & Sweller, J. (1999). A comparison of cognitive load associated with discovery learning and worked examples, *Journal of Educational Psychology, 91 (2)*, 334-341.

van Merrienboer, J .J. G., Kirschner, P. A., and Kester, L. (2003). Taking the load off a learner's mind: Instructional design for complex learning. *Educational Psychologist, 38 (1), 5-13*.

Yaman, M., Nerdel, C., and Bayrhuber, H. (2008). The effects of instructional support and learner interests when learning using computer simulations. *Computers & Education, 51*, 1784-1794.

5. AUTHOR BIOGRAPHY



John M. Bunch is President and Principle of The SAJES Group, Inc. He applies instructional science to solve business problems faced by corporate and other organizational clients, in addition to developing and delivering courseware in database and programming technologies. His research interests include the use of problem-based instruction in post-secondary career and technical education, and enhancing learner motivation through instructional design. He spent several years as a hands-on information systems professional and corporate trainer before receiving a Ph.D. in Instructional Technology from the University of South Florida.

Information Systems & Computing
Academic Professionals

EDSIG
Serving Information Systems Educators

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096