

Developing Software Simulations

Tom L. Hall
TCC Publishing, Inc.
Greenville, North Carolina, 25858, USA

Cathy W. Hall
Department of Psychology, East Carolina University
Greenville, North Carolina, 25848, USA

and

William W. Swart
Department of Marketing and Supply Chain Management, East Carolina University
Greenville, North Carolina, 25848, USA

ABSTRACT

Programs in education and business often require learners to develop and demonstrate competence in specified areas and then be able to effectively apply this knowledge. One method to aid in developing a skill set in these areas is through the use of software simulations. These simulations can be used for learner demonstrations of competencies in a specified course as well as a review of the basic skills at the beginning of subsequent courses. The first section of this paper discusses ToolBook, the software used to develop our software simulations. The second section discusses the process of developing software simulations. The third part discusses how we have used software simulations to assess student knowledge of research design by providing simulations that allow the student to practice using SPSS and Excel.

Keywords: Software Simulations, ToolBook, SPSS, Excel, Research Design.

INTRODUCTION TO TOOLBOOK

ToolBook Instructor 2004, a product from SumTotal Systems Inc. (www.sumtotalsystems.com), is a Windows authoring tool that allows content developers to create a large variety of different types of learning lessons including software simulations. Major features for developers include an Actions Editor and a Simulation Editor. Both of these features make use of a script-less programming environment where a large amount of interactivity can be built without the developer having to have much, if any, programming experience. The tool is designed so that anyone with good Windows skills can quickly create content with user interactivity, software simulations, and quizzes/exams.

Software application simulations can be developed where the learners can either view an animated demonstration of a series of steps used to perform a task, practice doing a series of steps (with feedback after each attempt), or go through a scored assessment of the performance of a series of steps. ToolBook can also be used to develop content-only learning lessons to

help teach the students the concepts related to the tasks that will be simulated. Likewise, ToolBook provides a powerful environment for developing exams that can be used to test the students' overall knowledge of the material presented. ToolBook provides flexible deployment options for publishing or deploying modular content to any standards-compliant Learning Management System (SCORM, etc.). The authoring environment of ToolBook provides an easy to use drag-and-drop programming environment for developing interactive content, software simulations, and exams. At the core of the drag-and-drop programming environment in ToolBook is the Actions Editor as seen in Figure 1.

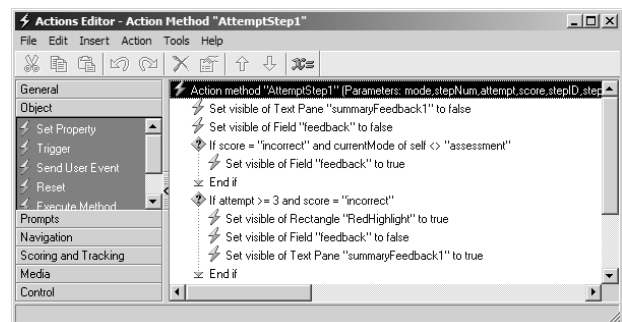


Figure 1
Actions Editor in ToolBook

This part of ToolBook provides categories of actions (programming code) that can be added to allow the developer to build any type of lesson. In Figure 1, the gray buttons on the left represent the categories that include:

- General
- Object
- Prompts
- Navigation
- Scoring and Tracking
- Media
- Control

In the example shown in Figure 1, the Object category is open and several Set Property actions have been added by drag-and-drop to control the visibility of a couple of objects on a screen. The Control category has been used to add an If/End If statement to check for conditions as a student does something on a screen. Other categories provide code for displaying popup alerts and queries (Prompts), code for controlling media (Media), code for controlling navigation with the lesson (Navigation), and code for scoring a lesson (Scoring and Tracking).

The various types of lessons that can be built with ToolBook include:

- Content – a lesson set up primarily to provide the student information on a concept. This type of lesson can include a variety of interactions such as clicking buttons and hotwords. A Content lesson can also include a variety of multimedia including audio and video. Additionally, this type of lesson can contain a variety of different types of practice or review questions that can be scored if desired.
- Exam or Quiz – a lesson set up to provide an assessment of the student's knowledge. An Exam or Quiz can include a variety question types including multiple choice, true/false, fill-in-the-blank, and drag and drop. A Score can be computed at the end and the exam results reported to a Learning Management System or posted to a database.
- Simulation – a lesson set up to provide the student practice in performing tasks for a software application. The simulation can be set up so that the student is shown the steps to perform a task (Demonstration mode). The simulation also can be set up so that the student is giving practice (Practice mode) where they have a defined number of attempts). Likewise, the simulation can be set up to provide an assessment (Assessment mode) where the student is given only a limited amount of help and only one attempt with the simulation being scored at the end in a manner similar to an Exam.

Key features of ToolBook include [1]:

Familiar user interface and built-in learning aids make it easy for teachers, students and staff to learn and use.

- Wizards, templates and a catalog of reusable objects help get the user up and running in a matter of hours, not weeks.
- Support for a wide range of media (Microsoft PowerPoint, PDF, audio, video, Macromedia Flash, Java Applets etc.) is provided that makes it easy to incorporate existing learning aids into online content.
- The lessons are deployable in a variety of ways – HTML, CD-ROM, network – with or without a learning management system.
- Advanced capabilities are there when the user is can incorporate scripting, software simulations and more.

ToolBook uses the book metaphor to describe the lesson that is created [2]. Each individual file or lesson is called a book. Each screen within the book is called a page. ToolBook is considered to be an object-oriented authoring environment, thus each individual element on the page is called an object. In the hierarchal structure of a ToolBook book, the page is an object, while the book is an object. Basic navigation between the pages of the book is then similar to how one might read a book, which is going from one page to the next. The big difference is that in addition to having a button that hyperlinks to the next page, there might also be a button that hyperlinks back one page, or even a menu button to return to a Menu or Table of Contents. In reality, buttons or hotwords can be used to hyperlink to any page within the book. Common objects on a page include buttons (such as pushbuttons, radio buttons or checkboxes), fields (to display text or allow the learner to type text), draw objects (lines, polygons), graphics (imported digital images), question objects, media players (for playing audio and video files), and simulation objects. In ToolBook, all objects have characteristics that we call properties. There are several different types of properties. Many properties relate to how an object looks. For instance a button has a border-style (such as pushbutton or radio button), a caption (the label that appears on the face of the button), a stroke color (the color of the caption), and a fill color (the background color of the button). These are all visual properties. Objects such as buttons can also have behavioral properties. For instance, a hyperlink can be added to a button that takes the learner to another page in the book when the button is clicked. The developer sets many of these properties by making selections from the available dialog boxes in a manner similar to what might be done in PowerPoint. As a software simulation is built on a page, all that the developer is doing is setting the various properties for the simulation object that appears on the page.

DEVELOPING SOFTWARE SIMULATIONS

As noted by Alessi and Trollip [3], computer-based tutorials traditionally consist of four key components: presentation, guidance, practice, and assessment. These four components can be adopted for developing interactive demonstrations and software simulations. Fleischer [4] suggests the following with respect to the key components of software simulation:

1. Presenting information. Start the user tutorial with a demonstration of the procedure. Using this "receptive" type of instruction, present how to complete the entire procedure without any user interaction. If the user has some experience, provide an option to skip the presentation or "test out" and move on to the next lesson.
2. Guiding the user. This phase is designed to guide the user step-by-step through a procedure by showing where to click or what to enter, and then prompt the user for input.
3. Practicing by the user. This phase is designed to help the user master the content of the course. Based on the learning objectives, this phase can include a lot of interactive exercises and minimal guidance. Rollover hints and audio can also be used to guide the user in the right direction.
4. Assessing the user's skill. This phase is designed to help evaluate the effectiveness of the training; for

example, by having the user complete a procedure without guidance. During this phase, guidance should be limited and interactions such as clicks, keyboard shortcuts and data input used.

As a developer begins to create software simulations for an application, it is important to consider that not all application functionality will be simulated. Estabrook [5] suggests that the developer start by looking at how frequent, critical, and difficult the various application functions are for the end-users, as well as what the functions have in common. Thus, the developer must make good decisions in how the simulations are identified and designed. Time and budget considerations must be taken into account. The main thing, as stated by Estabrook [5], is to “choose simulations wisely”.

The general steps for developing software simulations include:

- Identifying a key task to be simulated.
- Storyboarding the task and deciding how to break up the task into individual steps.
- Obtaining the screenshots (graphics) of the application being simulated.
- Developing the overall design of the final simulation product including background design, layout, and navigation with the product (the Starter application).
- Developing a Template with pre-programmed screens of the anticipated functionalities found in the application to be simulated.
- Using the Starter application and the Template along with the Storyboard to build the desired simulation.
- Testing the product built during several rounds of Quality Assurance with feedback provided via a Comment Collector System.
- Making any changes as suggested from the QA process.
- Publishing the final HTML product to be deployed to a Website or to a SCORM compliant Learning Management System.

An easy-to-use Simulation Editor in ToolBook allows non-programmers to quickly develop simulations in one or more of the available modes (Demonstration, Practice, or Assessment) with each simulation having one or more steps per screen. The Simulation Editor (see Figure 2) provides the developer options for automatically generating instructional text and feedback or the developer can enter their own instructional text and feedback. Each simulation can be set up so that an exercise is timed and a number of attempts can be defined and feedback given for each incorrect attempt. For instance, a step can be set up where the learner obtains very generic feedback text on their first or second incorrect attempt such as “Please read the instructions and try again.” On a third attempt the feedback can change to a direct instruction such as: “You should have clicked the OK button.” In this case, we can also use some form of highlighting to point to where the learner was supposed to click on the screen.

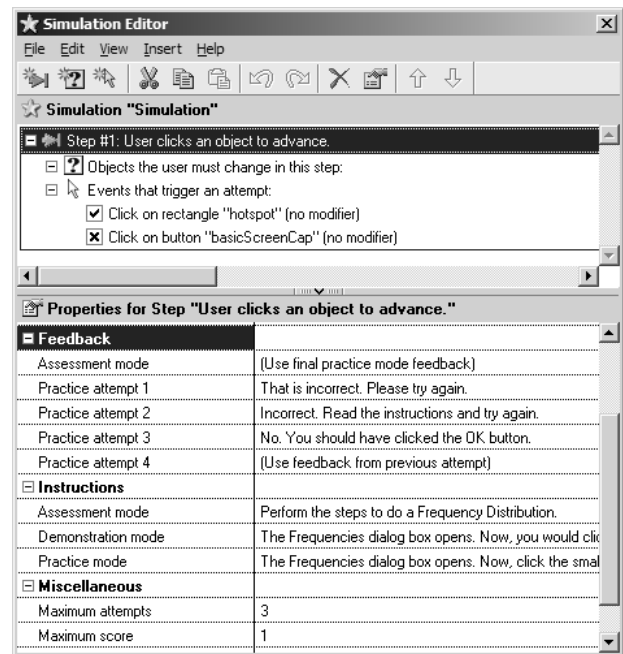


Figure 2
The Simulation Editor

In the example cited above, a red rectangle could be drawn around the button to be clicked and the final feedback text displayed in a popup text pane with an arrow pointing to the highlighted button.

As explained earlier, a simulation can be set up to run in any of the three modes (or all three within the same lesson):

- Demonstration
- Practice
- Assessment

Our current design incorporates all three of these modes by providing a Topic Mode Selection screen where the learner can choose either Practice or Assessment. A Show Me button on each simulation screen provides the learner the option of first viewing a demonstration of the steps to perform a task. Likewise, a learner can choose to practice first, and if they are unsuccessful, they can use the Show Me button to trigger the animated demonstration. A Try Again button is then available to allow them to practice again. In Demonstration Mode, animations are used to show the learner what to do for each step of the simulation. For the example of clicking an OK button, when the Show Me button is clicked, an animated cursor (mouse graphic) appears, moves to the area of the screenshot to be clicked (the OK button in this instance), and a sound (button click) is played. If the step is to click a field, type text, and then click a button or some other part of the screenshot, then the animated cursor moves to the field, a click sound is played, the text is typed one character at a time with a typewriter sound for each character typed, and then the animated cursor moves to what has to be clicked in order to continue. In Practice mode, the learner has three attempts and feedback is provided for each incorrect attempt. After three incorrect attempts, the learner is instructed to click a Continue button to advance to the next step. In Assessment mode there is only one attempt and

feedback provided immediately after an incorrect attempt. A score is computed at the end of the task (the series of simulation steps), and the data collected can be reported to a Learning Management System (LMS) if the product has been launched from an LMS.

CASE STUDIES

This presentation includes two case studies involving simulations. The first case study is how ToolBook was used to develop review lessons on research design and then software simulations for Psychology students. The second case study is how ToolBook was used to develop software simulations for a Business class to give students practice in using Excel.

Case 1

It is necessary for students entering upper level psychology courses to retain a working knowledge of research design as well as skills with basic applications involved in statistics. However, it is often necessary for the instructor to conduct an in-depth review of this information at the beginning of these courses before moving into course specifics. With the aid of simulations, it is possible to allow students to review the majority of this information independent of the instructor and to then have quick assessments with specified cut-off guidelines at the beginning of a course. This case study presents a series of lessons involving research design, basic statistical knowledge, data entry, and SPSS statistical analysis.

A Research Design lesson provides a quick overview of case studies, correlational designs, and experimental designs. Information is presented of developing research questions and then stating these questions as hypotheses. How hypotheses are stated often determine the research design that is needed. The strengths and limitations of each design are discussed. Dependent variables, independent variables, level of significance, and confounding variables are reviewed with respect to experimental design.

An Overview of SPSS lesson provides relevant screenshots of SPSS and examples of basic data entry and analysis of data.

The first simulation lesson, Entering Data into SPSS, involves entering a sample data set into SPSS. The data set is a very simple data set and is used to get the student accustomed to what is involved when data is to be entered.

The second simulation lesson, Select Concepts in SPSS, involves using SPSS to complete a basic statistical analysis. The student is walked through the process step-by-step using a predetermined data set to run a Frequencies Analysis.

Figure 3 shows a sample screen after the student has performed a task incorrectly three times. Note that a popup is displayed with very specific feedback text, an arrow pointing to where the student was supposed to have been clicked, and a rectangle is displayed around the part of the software application that was supposed to have been clicked.

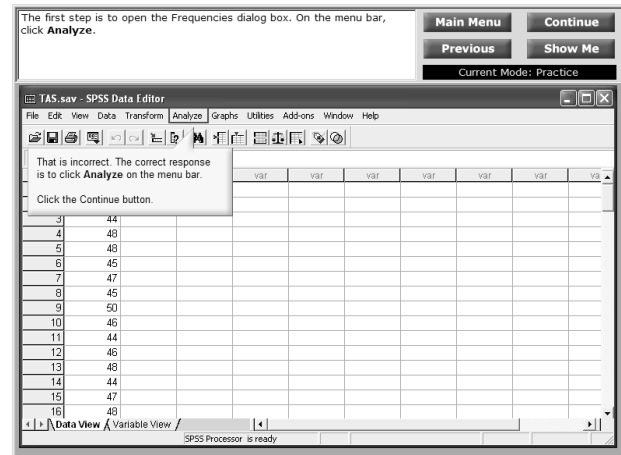


Figure 3
Simulation of SPSS

In the final lesson, students are given a short exam to assess their knowledge of research design and SPSS. The lesson consists of multiple choice and true/false questions and the student is given feedback for incorrect responses and a score is provided at the end of the exam.

This approach allows students to determine if they need to carefully review the information presented in the simulation or simply scan over prior to taking the exam or completing the SPSS project. The instructor is very aware that many students who may need to carefully review will not do so and opt to take the exam/project hoping to do well. Two separate exam options are given with the incentive to do well being that the best score out of two possible trials will be counted. This score will account for 20% of the students' first test score. This was determined to be a reasonable option given that in the past the instructor has included questions on research design and statistics as part of the first test and these questions have accounted for approximately 20% of the first test.

In addition, 49 students in the psychology course completed an opinion survey at the end of the semester in reference to the on-line simulation component of the course. Students were asked to respond to five questions in a four-point Likert format ranging from 4 denoting a positive response to 1 indicating a negative response. Participation in the survey was voluntary and no individually identifiable data was collected. The survey was approved by the university's Institutional Review Board and complied with APA ethical standards for research with human subjects. Results are presented in Table 1.

The majority of the students reported positive responses to the on-line simulation course component. Seventy-three percent of respondents reported the simulations were effective or very effective in presentation of material. Eighty-eight percent rated the ease in accessing simulations as easy to very easy. In responses to preference in having additional material presented through simulations, 59% reported simulations were preferred to much preferred, 27% reported it did not matter, and only 14% chose definitely not. For the majority of student

respondents (86%), the on-line simulations were either neutral or positive. Fifty-seven percent reported the simulation helped their grade on the on-line exam, 17% reported it did not make a difference, and 26% reported it hurt their grade. This was an interesting finding given that the mean score for the on-line portion of the exam was 17 out of 20 representing an 85% accuracy or mid-B grade whereas the mean score for those taking the in-class supplemental exam was 72% or low-C. Seventy-three percent reported being satisfied to very satisfied with the use of simulation with 27% indicating somewhat dissatisfied to very dissatisfied.

Table 1
Survey of On-Line Simulation Course Component (n=49)

Questions:	4 positive	3	2	1 negative
Effectiveness of simulations in presenting material covered on exam.	24%	49%	21%	6%
Ease in accessing on-line simulations.	48%	40%	10%	2%
Preference in having other material presented using on-line simulation.	21%	38%	27%	14%
Option of on-line simulation and exam helped grade on first exam.*	42%	15%	17%	26%
Satisfaction in use of simulation technology.	30%	43%	21%	6%

*only those students who took on-line exam responded to this questions (n=43 or 88% or respondents).

Results support the use of simulations as an effective means of presenting course material. It should be noted that in this course the simulations were a small portion of the overall course material and represented material that the students should have covered in previous courses. The main purpose was to review this material and then demonstrate competency via an on-line exam. The results may have been very different if a larger portion of the course was presented in this manner and if the material presented was “new” material. Further research in this

area is needed. The survey finding that 26% felt the simulation hurt their grade on the on-line exam was interesting given the high scores on the on-line exam. The survey was conducted at the end of the semester whereas the simulation and on-line exam were used at the beginning of the course. The delay in surveying students may have allowed additional factors to enter into how they chose to respond (i.e., not doing as well as they would have liked on subsequent exams, fatigue toward end of semester).

Overall, the results of the survey supported student acceptance and use of simulation technology as an effective and efficient means of presentation of course material. Achieving an optimal balance of simulation and standard course formats remains to be explored and should take into account the type of material being covered, the knowledge base of students, technological proficiency of instructor as well as students, and university support of technology.

Case 2

Two areas focused on in upper level business courses are providing students with a working knowledge of EXCEL and also providing skills with applications involving basic data analysis. The use of simulations can be an effective tool in meeting these objectives. Simulations allow the instructor to go beyond the presentation of the information in a classroom lecture format by actually having students apply the information in a controlled setting with immediate corrective feedback. Students do not have to work problem sets, wait for scoring and feedback from the instructor, and then try to correct any errors they may have made. With simulations the feedback is immediate and corrective learning can take place when an error is made. This following case study presents a series of lessons involving EXCEL commands and basic data analysis using EXCEL’s Chart Wizard.

The EXCEL lesson provides a quick overview of a data set representing the inventory of a business. The objectives of the lesson are to teach students how to count the number of occurrences of each outcome in a data set using the COUNTIF function, to construct a frequency table, and then to convert the frequency table into a pie chart using EXCEL’s Chart Wizard. A sample qualitative outcome is presented as part of the basic lesson. Upon completion of the lesson, the students are then asked to construct their own outcomes through the use of software simulations.

Figure 4 shows a screen from one of the EXCEL simulations. In this case, a single screen has two steps and the student has successfully performed the first step in Practice mode but has failed to perform the second step after three attempts. Feedback in the form of a popup note is provided along with an arrow pointing to where the student was supposed to have clicked on the screen in order to perform the next step correctly. A rectangle is drawn around that part of the screen indicating where the student was supposed to have clicked. Notice also that on this screen, information is provided in the form of a “Sticky Note” where the student is provided information related to an alternate method for performing the current task.

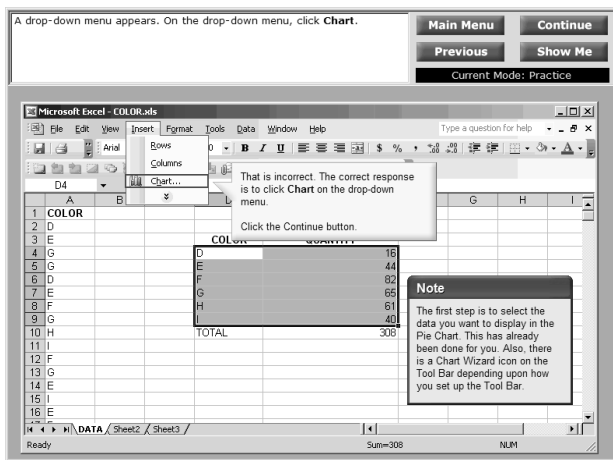


Figure 4
Simulation of EXCEL

In the final phase of the lesson, students are given a short exam to assess their knowledge of: 1) qualitative and quantitative data; 2) the use of the COUNTIF function to construct a frequency table; and, 3) their ability to correctly use the Chart Wizard to construct a pie chart. This includes testing their familiarity with each of the Chart Wizard dialog boxes that must be navigated before a final pie chart is produced.

SUMMARY

The above case studies provide examples of how ToolBook is used to develop software simulations in two academic areas: Psychology and Business. Several tasks in both SPSS and Excel are simulated.

The use of software simulation is an effective and efficient way to insure each student has a working knowledge of critical information. It allows students to work through problem sets with immediate feedback on their performance. The immediate feedback and “hands-on” learning provided by software simulations aid the student by allowing him or her to learn the material correctly and effectively the first time.

In summary, the use of software simulations via ToolBook allows for the effective presentation of information, provides guidance to the user, allows the user to practice the information presented to develop mastery of the content of the course, and provides for the option of assessing the user’s skills. As noted by Fleischer [4], these are the key components necessary in the development of software simulations.

REFERENCES

- [1] “You Have a Big Job To Do”, SumTotal Systems Website, <http://www.toolbook.com/education/index.php>, 2006.
- [2] Hall, T.L., An Introduction to ToolBook Instructor 2004, TCC Publishing, 2006.
- [3] Alessi, S.M. and Trollip, S.R. “Computer-based Instruction: Methods and Development”, Englewood Cliffs: Prentice-Hall, 1991.
- [4] Fleischer, S. “Creating Effective Software Simulations”, <http://www.adobe.com/devnet/robodemo/articles/simulation.html>.
- [5] Estabrook, S., “Making the Most of Software Simulations”, <http://www.learningcircuits.org/2004/sep2004/estabrook.htm>.