AUTHORING AN INTERACTIVE SAS COURSE

Ione H. Cockrell, North Carolina Educational Computing Service
Dorsey A. G. Condon, College of Charleston

The Computer Services Division of The University of South Carolina (CSD-USC) has used "An Introduction to the Statistical Analysis System", a computer assisted course written using IBM's Interactive Instruction System (IIS), to introduce users to SAS during the past two years. The course, called SASl for short, provides even the totally inexperienced computer user with the essentials to run relatively simple SAS programs in batch mode. The course consists of textual material interspersed with questions and is designed to test understanding. For a more detailed description of the course content, see "Conversational Instruction in the Statistical Analysis System" Glenn, D.A. and Cockrell, I.H., SAS Conference Proceedings, 1978.

This paper offers suggestions to persons considering developing a computer assisted course for new SAS users. Many of our suggestions are applicable to almost any computer assisted course.

The best approach to authoring an interactive SAS course will be determined by an installation's particular circumstances. It is not possible to construct a rigid set of rules which will work in every situation, but it is advantageous to extract as much as possible from the previous experiences of others. The basic guidelines which follow are a result of our experiences while authoring SASl.

Determine the Feasibility and Advisability of a Computer Assisted Course

A computer assisted approach to training users is not a good alternative for every organization. Before embarking on the time-consuming task of writing a computer assisted course, one should carefully weigh the resources required to develop such a product against the potential benefits. The prospective author should determine if the demand for the course is present and if the demand could be met more economically by other means. The availability of interactive terminals may be a limiting factor which will prevent some users from using computer assisted instruction.

Consideration should be given to the present and expected future workload of the computer system to determine if slow response times might result in student frustration and eventually a decline in the use of the course. A reliable and flexible computer assisted instruction package should be available before the project is begun.

At CSD-USC we felt that a computer assisted course was a viable solution to the special problems associated with training users in a large, geographically dispersed computer network supporting 50 batch terminals and 500 interactive terminals located throughout the state of South Carolina. In addition to the Columbia campus the network consists of 9 regional campuses, 5 other state supported colleges, 25 state agencies, and 15 technical education centers. The periodic short courses offered throughout the year obviously did not meet the needs of our out-of-town users. Other problems with traditional short courses are scheduling conflicts, the fixed pace of short courses, and the varying levels of experience of the participants. Because of the difficulties and inconveniences associated with traditional SAS short courses much staff time was being spent in one-to-one SAS teaching and consulting sessions. After due consideration, we decided to write a computer assisted course for new SAS users.

Generally, users have been very enthusiastic about their experience with SASl, but there have been some complaints. As the workload of the USC system has grown the response time has degraded. There is, of course, no simple answer to this user problem, but users have been encouraged to sign on at other than peak usage periods of the day. We have also found that some inexperienced users are frustrated at the onset by difficulties in connecting the terminal to the USC system. Because the CSD-USC users have so many different types of terminals, it was difficult to provide easy instruction, and hence some persons were discouraged from even beginning SASl.

We had hoped that little staff time would be needed for SASl after the initial development. However, as the popularity of SASl grew, we spent more time than had been anticipated attending to administrative duties such as
registering students and distributing supplementary reference guides. We also received a large number of calls from users who were having various types of minor difficulties with SASI. So a considerable amount of time will be required for maintenance and management of an interactive SAS course. This factor should be considered when determining the feasibility of such a course.

Define the Objectives and the Audience

Authoring an interactive course is no different from authoring any other course in this respect. A clear definition of the course objectives and a keen understanding of the audience are vital to the endeavor's success. During this process, it is helpful to consult with any group of possible users concerning their needs and capabilities. A thorough understanding of the audience and course objectives will maintain the proper tone and content of the project through all phases of design and development.

The course objectives will be largely determined by the needs and characteristics of the audience. It will not be possible to meet every user's specific needs without degrading its overall usefulness. It is a good idea, if possible, to keep the course rather limited in scope. An overabundance of detail is boring, and frequently overshadows the primary focus and flow of a course. For example, the thrust of SASI is not to replace the SAS User's Guide, but rather to prepare the inexperienced user to use the manual in a productive fashion.

Develop a Plan

A good plan, constructed during the early stages, will contribute both to a usable end product, and to minimum confusion and problems during development. The plan will be strongly influenced by your preliminary work in defining the audience and objectives. Major considerations include the following:

a. Number and length of sections. The modular approach with many relatively short sections is generally preferable to a few extensive sections. Sections that are too long can become very tiring for users and frequently force each user to sit through every topic in the course. Additionally, the modular approach offers several important advantages to the author. More than one individual can be involved simultaneously in the authoring and entering of data. Such a course is also easier to update and expand. It is important to consider only approaches which offer considerable flexibility both to the student in taking the course, and to the author in writing the material, adding to the course content and modifying the text.

b. The format for the tutorial sections. Many different forms for the tutorial sections are possible, but we had considerable success with the one which consisted primarily of short segments of text interspersed with questions. Typically, each section of text presented one fundamental concept. Frequent questioning lets the user know how he is progressing and maintains his interest.

In computer assisted instruction, much of what the student learns is from the mistakes he makes in responding to questions. A substantial amount of the time which was required to write SASI was spent in designing questions to test a student's understanding of the concepts and in anticipating and responding to incorrect responses. In SASI, several types of questions are employed including short answer, multiple choice, and true-false. Whenever appropriate, the student is given several opportunities to respond with a correct answer. If the student types an anticipated incorrect response, an explanation of why it is incorrect is given and he is usually asked to try again. If the wrong answer is not one that is anticipated, he is merely told that it is not correct and asked to try again. If after several attempts, the student has still failed to give a correct response, he is given the correct answer with an explanation. The student never leaves a question until either he types the correct response or the correct response is presented to him. The student is never ridiculed for an
incorrect response. Using IIS, the author is able to code in several correct responses as well as to allow certain variations in spelling. For example, if the correct answer to a question is "TRUE," the author may code so that either "YES," "RIGHT" or any word beginning with the letter "T" would be considered correct. Another IIS feature allows the student to request a hint whenever he feels he needs assistance in answering a question. So if a hint has been coded for the question, the hint appears on the terminal.

An interactive course should be designed so that the experienced user can skip certain parts of the text and the novice can repeat sections when necessary. In SASL, this is accomplished by preceding each section with a summary of objectives and giving the user the opportunity to proceed with the section or skip it entirely. Similarly, a capsule of objectives and a summary is given at the end of each section. The user is given the opportunity to proceed to the next section or return and repeat the tutorial portion of the current section.

c. The specific information to be contained in each section. Determining the specific information which will be contained in each section is one of the most critically important phases of development. It is suggested that you define each fact to be presented and each task to be accomplished within each section before beginning to write the section. Again we point out that the user may be bored by unnecessary detail, but on the other hand he needs to have all the facts necessary to achieve mastery of the material. In any case, the course should do more than present facts. It should teach the user how to accomplish tasks. A mere potpourri of facts is of little utility.

d. Use of a supplementary reference. A supplementary reference guide can be helpful for the user as well as the author. We found that such a guide eliminates extensive note-taking by the student and provides a ready source of reference material such as general rules, general forms, examples, etc. It also allows the student to have an extensive discussion of a problem in hand while answering questions about it at a terminal. Frequently, all the information would not fit simultaneously on a screen, thus requiring the student to return to display the information again and again. The guide is also a convenient method of giving sign-on instructions and other preliminary instructions. It is preferable if the reference guide can be made available in a machine-readable form to facilitate distributing and updating. In this way the reference guide and course can be kept in parallel. Additionally, the distribution of the guide is much easier since any user, regardless of his location, can obtain a copy of the guide merely by submitting the proper Job Control Language.

Finalize Your Plan

After the details of the format and content of the course have been developed, it is wise to ask several interested parties to provide some constructive criticism. The individuals must be knowledgeable, experienced, and interested in order for this step to be helpful. They must understand your audience as well as what you are attempting to convey to users. Another consultant or a faculty member who has taught SAS is a candidate in an academic environment. Careful consideration of such criticism can be extremely helpful in developing a highly usable product.

Gain a Thorough Understanding of the Authoring Software

Understanding the authoring medium may be, depending upon its complexity, one of the most time consuming and difficult phases of developing your interactive course. For this reason, you should determine during the early planning stages how much, if any, assistance you can expect to obtain from others. There may be consulting services available from within your organization or from the software distributor. Programmed texts, workshops, CAI courses, or introductory guides may be available through the
distributor for persons who are interested in authoring such courses. The distributor may be able to provide you with names of persons either locally or at some other location, who have attempted similar ventures. Such contacts might prove helpful by sharing experiences, by assisting in trouble-shooting when problems arise, or by giving advice concerning approaches to take, software features which work well or poorly, etc. Learning to use IIS required a significant investment of our time, although having a thorough understanding of the features of IIS undoubtedly saved time and work in the long run. Fortunately, IBM supplies an IIS course to instruct users in authoring courses. This not only provided us with the details of authoring techniques but also with a good example of an IIS course. From the course and a user's manual we obtained most of the information we required to begin designing and authoring SAS1.

Write a Mini-Course to Test Features

If you have not used the authoring package previously, the only feasible way to be sure that all key software features will work to your satisfaction is to author a scaled-down version of a course under the system. It is advisable to use every possible option that you are considering, such as pretests, branching, answer analysis, and posttests. Every type of question (short answer, multiple choice, true-false, matching, etc.) should also be included in the mini-course.

The mini-course should be tested on each terminal type that will be used by students. A certain approach to spacing may look fine, for example, on a CRT, but be totally unacceptable at a hardcopy terminal. The sign-on instructions also may be vastly different from device to device. These are important considerations when entering the course and providing instructions to prospective users.

We extensively tested the features by signing on as a student and utilizing as many options as possible. In this way, we were able to eliminate many potentially serious problems as soon as they arose. This might appear to be an unnecessary step, but it is one of the most critical. During this phase of development, we discovered many features of IIS which were not operating as advertised (some of which were due to the implementation of IIS at our installation).

Write and Enter Course Material

At this point the details of what the course will contain have been established, but the material has not been finalized and put in a machine-readable form. If the earlier stages of development have been carefully and thoughtfully completed, writing the course material should prove to be relatively easy.

When you reach the entry stage of development, enlist clerical help if at all possible. This may require some training especially if the system prompts for input. However, relieving the author of this time-consuming task will free him for the all-important task of editing, testing and debugging the course.

Test and Debug the Course

After the course has been entered and edited, sign-on as a student and test each option of the course as thoroughly as possible. Attempt to give a wide range of correct and incorrect answers. If branching options have been included, test them at various stages of the course to insure that they work properly.

After you are satisfied with the content and performance of the course, ask staff members or interested faculty members to take the course. Emphasize that the course is in development and that you are interested in constructive criticism.

IIS, as well as other authoring systems, allows the author to collect data on student responses to questions. This aids the authors in identification of questions which have been entered incorrectly or worded poorly. Unfortunately, we did not use the feature very often although we do recommend it. Also, through IIS, the student should be able to use his terminal to send messages and suggestions directly to the authors. Although we did not successfully utilize this feature, we feel that it should provide the author with valuable feedback from the student during the early phases of use.

Conclusion

We feel that the computer assisted course to teach SAS to new users has proven to be a good alternative to short
courses at CSD-USC. During the past two years, several hundred users have taken SASI. It has been used by whole classes of students as well as new CSD-USC staff members. Currently, approximately 170 students are signed up for the course. We are certain that much staff time has been saved, that our users have benefited from their participation in SASI, and that SAS usage has increased significantly as a result of using this technique.

References

