Teaching the Mechanics of SAS with a Self-instructional Manual: A Case Study

Barbara M. Florini
and
Jack Six
Center for Instructional Development
115 College Place
Syracuse University
Syracuse, New York;

Abstract. With the increasing demand for skill and knowledge in the operation and application of computer technology, colleges and universities are being pressured by their students to provide effective and efficient instructional programs in this area. The following case study describes how one university, through its instructional development agency, was able to develop, evaluate, and refine a self-instructional manual that provides management students with the basic skills required for using the Statistical Analysis System (SAS), a major statistical program available for mainframe computers. The successful use of this manual with several hundred students over the course of five semesters and three summer sessions suggests that the manual may be a useful model for institutions with similar needs.

This paper describes the development, use and evaluation of a self-instructional manual to help solve a persistent computer-related problem at a university. Each semester large numbers of students from various disciplines are required to develop basic skills in using the Statistical Analysis System (SAS), a statistical analysis and reporting program that is available on the campus mainframe computer. Because of the wide availability of SAS on large computing systems, it is unlikely that the problems students face in learning to use SAS are unique to one campus. This case study is presented in the hope that others might benefit from the experience described below.

Following a description of the initial situation, the following topics are discussed:

- the development process leading to the manual.
- the manual itself.
- the introduction of the manual into the instruction setting.
- the evaluation plan and the results of the evaluation.

The paper concludes with several general observations and recommendations.

Description of the Situation

For some time, the use of SAS by students has taxed the academic computing consultant resources on campus. The general feeling was that students new to computing were having difficulty learning to use sophisticated applications programs like SAS in spite of classroom instruction and the availability of a consultant's room in the computing center. As a result, the Center for Instructional Development at the University was asked by a senior administrator with oversight for campus computing services to devise some efficient and effective means to help students learn SAS. Given this request, the growing demand on computing resources, and the large numbers of students needing to learn SAS, the project was given high priority within the Center.

Problem Conformation

In an effort to find out if the problem was amenable to an instructional solution, a series of discussions was initiated with members of the computing center staff and faculty having prior experience teaching SAS. These discussions confirmed that large numbers of students routinely sought help in running even simple SAS jobs despite classroom instruction, ready availability of reference documents, and on-line instructions for using SAS.

Problem Analysis

The next step was to identify the major elements of the problem. This was done through discussions with representatives from the Academic Computing Center, faculty members, and students. The following five major factors emerged.

1. SAS Itself. The Statistical Analysis System is a powerful, complex computer program. It enables users to analyze data by means of many different kinds of statistical procedures and to generate data reports in various ways. Although it serves the needs of people in many disciplines and staff positions, the experiences of a large number of users suggest that SAS is not especially accommodating for beginners. Preparing information for use by SAS is akin to programming. SAS has a language of its own that is not intuitively obvious, and has strict rules for entering information.

For example, to enter variables in a SAS program, new management students have to recognize the relationship between the data on a set of sales receipts and the way in which that data must be organized for use by SAS. Because SAS does not permit spaces within variable names or variable names in excess of eight characters, students have to rename variables like order number, sales region, and product number to a format SAS can recognize: ORD · NO, REGION$, PROD · NO$. (The $ symbol indicates that the values stored in the variable will consist of characters or a mix of characters and numbers rather than numbers alone.) Students must then know to precede the list of variable names with INPUT and conclude the list with a semicolon.

2. Hardware and Related Matters. Two aspects of the hardware factor emerged. The first concerns the many different kinds of terminals on campus. They vary in appearance, location of switches and control knobs, and keyboard design and labelling. Experienced

users scarcely notice the differences, but beginners are confused by them. The second aspect concerns the mainframe. Beginners are often perplexed by the complex operating system they must use in order to run a SAS program. They are also disconcerted by computer system failures or crashes, whether or not they are forewarned.

3. Diverse Instructional Efforts and Expectations. On examination, a diversity of instructional materials available to students for learning SAS in both nonclassroom and classroom settings was found. These materials generally assumed that all students had identical learning needs. However, instructors in different schools and colleges wanted students to learn SAS for various applications. Some instructors focused on statistical analysis, and others on report generation. In addition, instructors often differed in their expectations as to the level of expertise beginning users needed to achieve. Because of the diverse efforts in both classroom and nonclassroom settings, students lack common fundamental skills in the use of SAS. Therefore, instructors planning SAS-related activities at all levels face the recurring dilemma of where to begin the lessons.

4. Demands on Academic Computing Center Staff. In part, the role of the Academic Computing Center Staff, along with its student consultants, is to assist members of the university community with using the mainframe computers. The staff felt that a disproportionate amount of this service time, however, was consumed by questions dealing with elementary applications of SAS.

Discussion

On examining the above factors, it was obvious that there were some things about which nothing could be done. For example, it would be unrealistic to recommend substitution of a more accommodating mainframe program in place of SAS. Because of its overall quality and power, SAS had recently been selected as the statistical and report program to receive full Academic Computing Center support for the foreseeable future. Moreover, it was not possible to provide campus-wide standardized hardware, nor to make SAS itself or the mainframe operating system easier to use. It also was not possible to impose a common SAS instructional policy throughout the university and its schools and colleges. Finally, nothing could be done to increase Academic Computing Center staff available to help students having difficulty with SAS. It was equally clear, nonetheless, that some changes could be made by approaching the SAS problem from an instructional perspective.

At the same time, it was felt that any attempt to deal with the SAS problem should:

- benefit a large number of students,
- reduce instructors' burdens,
- withstand obsolescence,
- be both efficient and cost effective, and
- provide a common foundation for advanced SAS work.

The goal of the CID staff was to develop a means for teaching SAS that would offer maximum benefit to all interested parties—administrators, faculty, students, and Academic Computing Center staff. The staff began to consider the possibility of instruction that could be readily tailored to meet needs identified in specific courses.

ing SAS. One of these professors, on behalf of his department, was about to revise such a course in Management Information Systems with the assistance of CID staff. He had received an award under a Lilly Endowment Grant to the University to help support this work. Since it was clear from preliminary discussions with the faculty member that SAS instruction would continue to be a major component in the course, there was mutual agreement to make the SAS project part of the course re-design effort. The only qualification was that the new SAS design be ready in four months for use at the beginning of the next semester. Other course changes would be introduced gradually.

Course Context

Previously, Center staff had been extensively involved with departmental faculty members in redesigning the entire Management Information Systems curriculum. One outgrowth of the curriculum redesign process was the proposed modification of several courses, including the introductory Management Information Systems 255 (MIS 255) course. The course enrolls several hundred students each year, and it serves diverse audiences. In addition to being a required course for management students, it is also chosen as an elective by students from other schools and colleges in the university.

Users have reported a high level of success and satisfaction with the manual. The manual reduces amount of time required to teach SAS and provides students with a common foundation for future use.

The Development Process

The instructional development process followed in the project is based on the standard CID model (Diamond, Eickmann, Kelly, Holloway, Vickery & Pascarelly, 1975). As applied in this instance, the process consisted of: (a) clarifying the problem, (b) identifying the desired SAS content, (c) determining an instructional flow, and (d) selecting an appropriate means for delivering the instruction, which in this instance turned out to be a self-instructional manual.

Preliminary Activity

Before beginning the development process, however, it was necessary to find a faculty member willing to participate in the development of an instructional program for teaching SAS. Several faculty members were teaching courses that included instruction in us-

After several instructional development sessions with the instructor, the following objectives were formulated for the SAS unit.

On completion of this component, all students should be able to:

- organize given data into a form usable by the SAS program,
- enter the organized data into the computer,
- produce computer-printed reports based on this data by using appropriate SAS procedures.

Following the formulation of objectives,

attention turned to identifying the instructional content and its sequence for the unit.

SAS Content

The professor provided a sample of the kind of SAS-generated report he wanted students to produce. From this, it was relatively simple to identify those procedures of SAS that students in his course needed to learn. These included: Print, Sort, Link, and Merge. The next stage was to identify the sequence of steps students must follow in order to produce reports using these procedures.

Task Analysis

The correct sequence of steps was determined by working backwards from the desired final report the students were to submit. The questions were: (a) How was this particular stage of the report reached? and (b) What specific steps must be taken to reach it? Two distinct sets of activities emerged as illustrated by Figure 1.

Pre-SAS Activities

The results of the task analysis revealed that, in order to meet the needs of entry level students, the SAS instructional sequence had to begin with learning to (a) use a terminal and keyboard, and (b) interact with non-SAS aspects of the computer. These include learning to take appropriate action in the event the computer ceases to operate while in use as well as simply learning to sign on and off the computer.

The task analysis revealed another part of the sequence students must learn before actually using SAS. Few students readily see the relationship between pieces of information on a set of documents, and the need to organize these pieces systematically before entering the information into the computer. If this organizing activity, known as preparing the data step, is ignored, SAS cannot process the information as desired.

Finally, because entry errors are inevitable and revisions of data are often necessary, students need to learn basic editing techniques. Editing the information entered into the computer is, therefore, the last part of the pre-SAS sequence.

SAS Activities.

The components in this sequence introduce students to those features of SAS that enable them to create the kind of reports desired by the professor. After the data is edited, students can run their first SAS jobs. Despite efforts to ensure that their input is free of errors, it is a near certainty that some students' jobs will not run properly. Typing errors, omission of critical symbols, or the incorrect choice of SAS commands are common reasons for the failure. Until students have error-free data input, they cannot obtain the required reports. It is essential therefore, that students learn what steps to take when their SAS jobs do not run properly.

The task analysis also helped reveal the specific content that would need to be taught in order for students to be able to produce the required reports. It was clear that some new concepts must be introduced. Without the combination of skills and conceptual understanding, students would lack a common foundation for advanced work in SAS, an important project objective. The next step therefore was to identify any other relevant content such as nomenclature and concepts necessary for students to be able to use SAS for Management Information Systems applications.

> The goal was to develop a means for teaching SAS that would offer maximum benefit to all interested parties.

Associated Concepts

Identifying the procedures students must follow in order to produce reports using SAS was straightforward. Identifying the relevant concepts needed to understand the reasons underlying the procedures was more challenging. There are many concepts related to SAS. Criteria for selecting which of them to include in the instruction were also derived from the task analysis. The decision was to include only those concepts highly related to the immediate pre-SAS activities.

Concepts chosen for special emphasis included: file, permanent and temporary storage, workspace, SAS syntax, and computer crash. Crash was selected because of the unnerving effect computer crashes have on new users; SAS syntax because its correct use is necessary for correct program execution. The remaining three highly abstract concepts were chosen for special emphasis because all evidence suggested that they are: (a) persistently difficult concepts for students to comprehend and (b) essential concepts for students to understand in order to become self-reliant users of SAS.

Selection of the Delivery System

After the procedural and conceptual content were identified, attention turned to determining the most efficient and effective delivery system in relation to the following criteria:

- reduce burden on instructor,
- minimize costs,
- ensure common learning outcomes,
- facilitate learning for students.

The first three criteria, as indicated earlier, evolved from discussions with all the parties interested in the project. The fourth criteria reflects a desire to reduce barriers to student learning and to minimize sources of student frustration when learning to use SAS. Given these criteria, the previously identified content, and the specific instructional setting, a variety of instructional methods and delivery systems were considered before deciding to produce a self-instructional manual.

The Manual

The content and structure of the manual were determined by the previously discussed objectives and task analysis. The presentation of the content was heavily influenced by the work of Keller and his associates (Keller, 1983) who devised a model to assist in designing instruction that is motivating. Although description of the model is beyond the scope of this paper, its application is reflected in the manual by the use of (a) attention-getting devices, (b) relevant examples, (c) tangible evidence of success in the form of frequent printouts, (d) learner management of the pace of the instruction (e) and analogies. Expectancy for success is also fostered. Other features used in the manual are characteristic of systematically designed instruction. These include: varied presentation forms for the complex concepts; graphic representations of on-screen information; chapter

overviews and summaries; self-tests; and examples of correct printouts.

Process for Writing the Manual

Using the results of the task analysis (see Figure 1), the professor provided an initial draft of the procedural steps for the first unit. In order to check each procedure, the development staff entered the information into the computer and attempted to get the desired output. Based on this experience, the procedural steps were clarified through discussion with the instructor.

Simultaneously, the staff discussed how to incorporate the necessary concepts into the procedural steps. The scope of coverage for any one concept was deliberately limited. The intent was just to give students sufficient information so that they would have a basic understanding of what they were doing and have a sound basis for continued concept formation.

An additional concern at this same time was to develop appropriate examples to give relevance to the procedural and conceptual information. Since the majority of the students who would be using the manual were from the School of Management, examples were drawn from the business world. Attention to all these concerns was reflected in the manual in the very first draft, if only by rough notes such as "elaboration of file concept goes here." The process just described was reiterated for all six units in the manual.

Product Description

The manual is a bound seventy-nine page document (8 ½ x 11) containing six units. All units begin with a short overview telling students what they are about to do. The bulk of each unit leads students through the steps necessary to accomplish a pre-determined outcome. Emphasis is on following correct procedures. Concepts are integrated into the procedures where appropriate, and elaborated, when necessary, at the end of the unit.

Because relevant management examples were considered important, six order forms, containing seven variables, were created (see Figure 2). These provide the data used throughout the manual. The variables are: customer number, date, salesperson, order number, sales region, product number, and quantity ordered.

Correct naming of each variable is introduced early in the instruction. The difference between string and numeric variables is discussed along with the naming conventions. Also, the distinction between an observation, in this case one order form consisting of several variables, and a variable is illustrated. Finally, students are helped to organize the information into a data matrix (see Figure 3). Students could now enter the

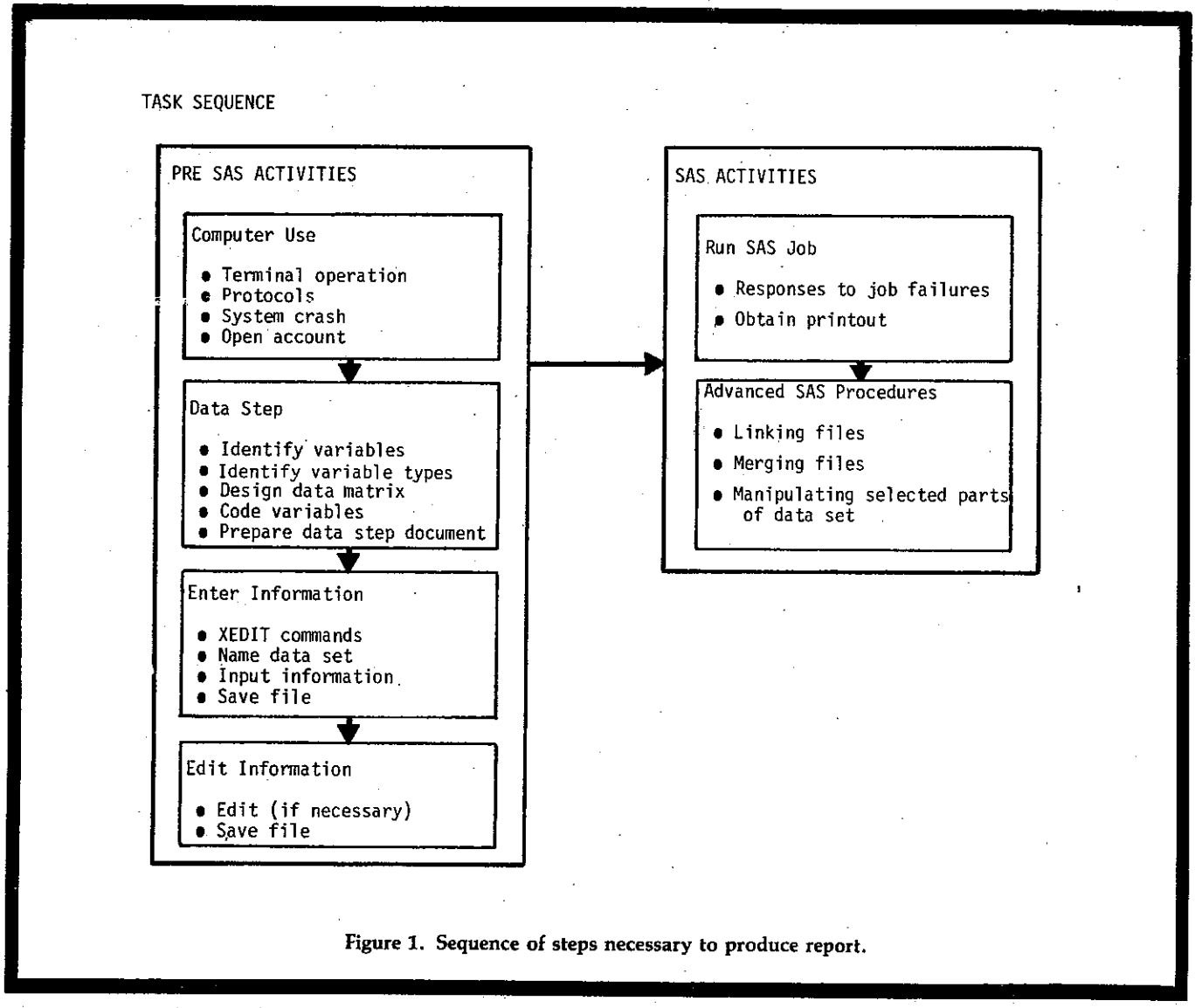

Figure 1. Sequence of steps necessary to produce report.

data into the computer. In the remainder of the manual, students learn to enter their data, revise it, and generate assorted reports through a series of exercises.

Each unit ends with a brief summary review. A crash sheet is printed on colored paper for quick reference and bound in the appendix section. A review of the logon procedure, a trouble shooting guide, short bibliography, and a summary of the computer commands used in the manual conclude the appendix section.

Formative Evaluation

When all the units were completed, five people not involved in the project and having little, if any, familiarity with SAS and the mainframe computer volunteered to participate in a one-on-one tryout of the first completed draft. Each of the five formative evaluation tryouts was conducted successively. Thus, the manual was refined during this process and all succeeding volunteers benefited from each predecessor's experience. The final manuscript of the manual was then prepared and printed for use (Tanniru, Florini, Six & Williams, 1985). Figure 4 illustrates the flow and content of the manual.

Implementation

The manual was used initially in the 1984 spring semester. The original intent was to use it in one or two sections of the introductory course in Management Information Systems for field test purposes. Following revisions based on this field test, the manual would then be used in all sections of the course in the next semester. Once the manual was produced, however, several colleagues of the course instructor decided to use the manual in advanced courses. Because these instructors were not familiar with the manual and its intended use, a teacher's guide was prepared for them.

Evaluation

The evaluation had a threefold purpose: (a) to test assumptions made about the computer skills of the intended users, (b) to identify any systematic differences within segments of the intended user population, and (c) to obtain feedback from both instructors and student users about the manual. Based on this information, revisions would be made to the manual in time for its use in all sections of the course for the 1984 fall semester.

Evaluation data were collected from students and instructors in both the advanced and introductory courses. Keeping in mind the intended user audience, however, particular attention was paid to the evaluation data obtained from the students and instructors in the introductory sections. The data from this smaller audience was used for assessing the user's computer skills and individual differences among users.

Data Gathering

The data were gathered by means of an interview protocol and a questionnaire. The protocol was designed for use with all instructors employing the manual in their courses. This procedure permitted the staff to get their feedback on a uniform set of questions and to probe for additional comments on the manual. A single experienced interviewer, not previously associated with the project, met individually with each instructor to help insure consistency in data collection.

A two-part questionnaire was designed for use with the students. The first part contained 32 items. The first five items asked for demographic data. The next eight elicited the students' prior experience, if any, with computers and with SAS. The remaining nineteen items covered: (a) the students' experiences while using the manual, (b) perceptions of the effectiveness of the six individual

ABC COMPANY
Sales Order Form

Customer No. ST0562 Date: 080382

Salesperson: Smith Order No. 242

Sales Region: NE X SE __ NW __ SW __ MW __

Product Number	Quantity Ordered
X17	40

Figure 2. Order form.

Figure 3. Data matrix.

	SEVEN VARIABLES					
CUST_NO$ Customer Number	Date Date	S_NAME$ Salesperson	ORD_NO Order Number	REGION$ Sales Region	PROD_NO$ Product Number	QTY_NO Quantity Ordered
ST0562	080384	SMITH	242	NE	X10	50
ST0562	080384	SMITH	242	NE	X17	40
ST0562	080384	SMITH	242	NE	X19	35
TAN156	051884	JONES	784	MW	X12	45
TAN156	051884	JONES	784	MW	X17	82
TAN156	051884	JONES	784	MW	X19	25

units, and (c) the students' levels of confidence in their abilities to produce reports using SAS. The second part of the student questionnaire had five open-ended questions. Two questions checked the implementation and use of the manual. The other three asked each student to identify the best and worst features of the manual and to make recommendations for improving it.

Instructors were asked to have students complete the questionnaire during class time. No attempt was made to obtain responses from those absent when the questionnaire was administered, and some students did not take time to respond to questionnaires that were distributed near the end of the class period. Due to examination schedules, one class had to be omitted from the evaluation. A total of 85 students responded, 44 of whom were in the MIS 255 target group.

Results

Students. Analysis of the data generally confirmed the assumption made about the intended user population. That is, they had little prior experience in using either the mainframe computer or microcomputers for word processing, data analysis, or programming. The analysis also indicated that there were no systematic differences between: full-time or part-time students, males or females, various age groups, and those enrolled in the main campus or the university's evening school.

When asked to state the best feature of the manual, most students cited its simplicity, step-by-step guidance, and clarity of instructions. Eleven percent of the students did not respond to this open-ended question. Other favorable student comments mentioned the constant review of previous units, the helpfulness of graphics, flexible and efficient use of student time, and the examples.

On the other hand, in response to the question, "What was the worst feature of the manual?" forty-one percent of the students either wrote "none" or did not respond. Among the students who did reply to the question, there was no consensus regarding a worst feature. For example, two students mentioned the redundancy of some procedural steps as a worst feature. Four cited the cost of the manual ($6.50 for 79 pages). Two students said the manual was too helpful in that they did not have to think much about what they were doing. Several students used the space allocated for this question to suggest that the scope of the manual be expanded.

Instructors. When asked to identify the strengths of the manual, the instructors cited the simplicity of explanations, heavy provision for hands-on experience, the quality of the writing, the visualization of files, the way in which the true beginner was helped, and the general approach to the mechanics of using the SAS package. Based on the feedback from the instructors, there were no systematic weaknesses in the manual. Several, however, wished it had gone farther in scope and complexity. One said it bordered on the too-simple.

Based upon the evaluation and on consultation with the cooperating professor, no substantive changes were made in the SAS manual. Self-tests were added for each unit as well as a comprehensive final self-test to help students think about what they were doing. Also, a reproduction of the final merged SAS program was appended to Unit 6. This made it more convenient for students to check their work by looking at a single document rather than by paging through the unit for the program segments.

Conclusions and Recommendations

As of this writing, several hundred students in the School of Management have used the manual. An instructor in the School of Information Studies also chose to have students in one of her courses use it even thought all examples are geared to management students. Generally, users have reported a high level of success and satisfaction with the manual. According to the instructor in charge of the management course for which the manual was created, it does reduce the amount of time required to teach SAS in that class and also provides students with a common foundation for future instruction in the use of SAS.

The findings suggest that novice computer users can effectively be taught to interact correctly with a major program on a mainframe computer through the use of a self-paced, self-teaching manual that requires minimal support from instructors. Although the content of the manual deals only with a portion of the SAS program, confidence that the instructional approach described here would be effective with all components of SAS or any other statistical program seems justified.

In order for such a manual to be effective, it is important that there be an exact match between the equipment and software used by students and between the description and operation of these items as presented in the manual. Moreover, it is essential to monitor any changes in the computing system and revise the manual accordingly. Some computing centers frequently change

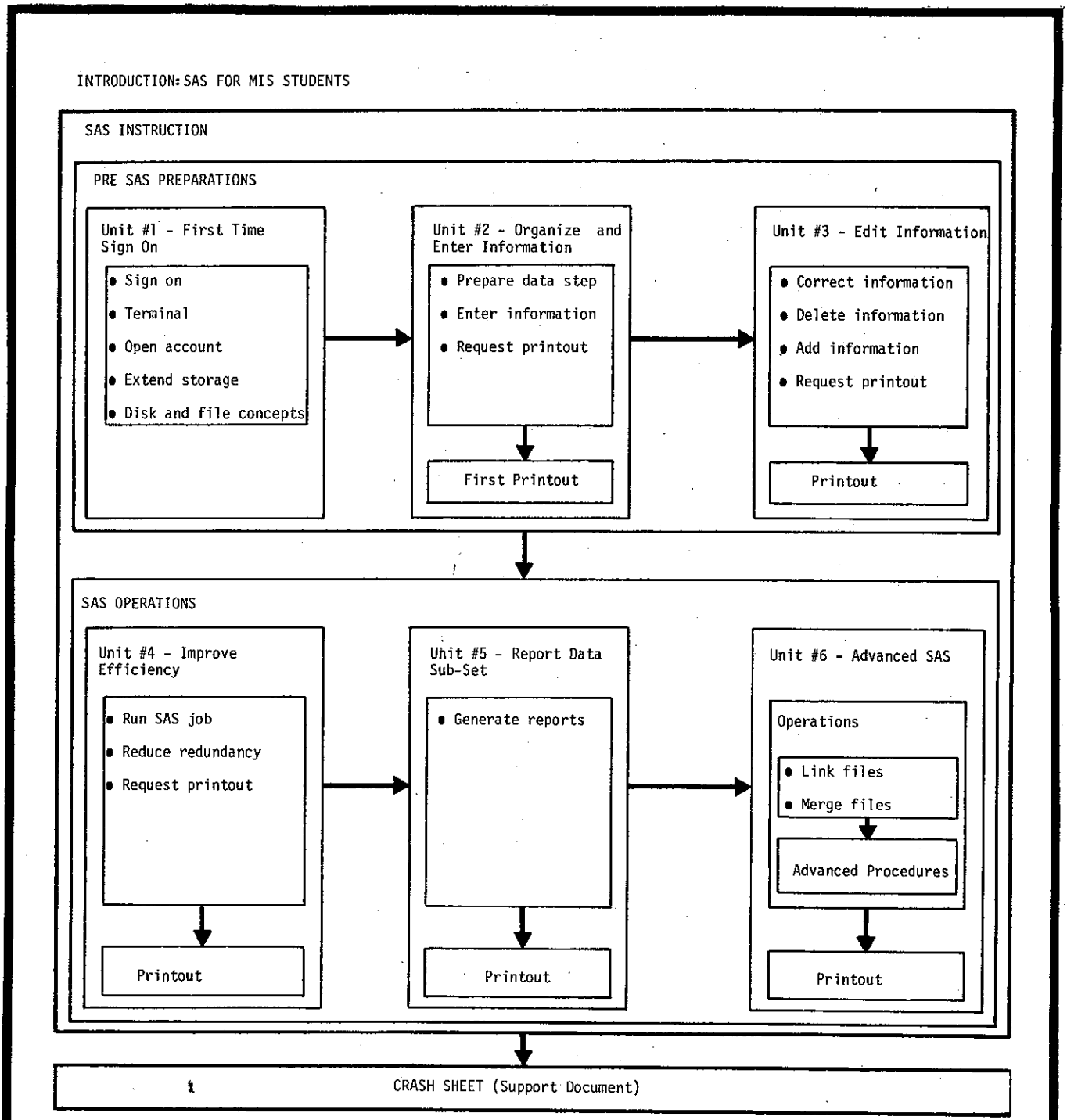

Figure 4. Blueprint of the manual.

various features of their overall systems. It is equally important to monitor changes in the user population since it is quite likely that more people will be arriving at college with fairly sophisticated computing skills.

Finally, someone needs to accept responsibility for keeping the manual up-to-date by monitoring both the computing system and the student population for changes. This person needs to be identified from the beginning of the project and to become a working member of the project team. Without this level of commitment and involvement, the person is unlikely to understand the cohesive nature of the instruction and, therefore, is unlikely to be able to maintain the instructional integrity of the manual.

References

Diamond, R. M., Eickman, P. E., Kelly, E. F., Holloway, R. E., Vickery, T. R., & Pascarelly, E. T. (1975). *Instructional Development for Individualized Learning in Higher Education.* Englewood Cliffs, New Jersey: Educational Technology Publications.

Keller, J. M. (1983). Motivational Design of Instructions. In C.M. Reigeluth (Ed.), *Instructional-Design Theories and Models: An Overview of Their Current Status* (pp. 383-434). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Tanniru, M., Florini, B., Six, J., & Williams, D. (1985). *Mechanics of SAS: An Introduction for Management Students.* (3rd rev. ed.). Syracuse, New York: Center for Instructional Development.