INTRODUCTION

In SAS training courses, as in most other courses, a student really begins to learn the material when he is faced with an application problem that needs solving, not while he is listening to the lecture. If, for one reason or another, his lecture notes are insufficient to help him solve his application problem, he needs a handout or manual that will allow him to locate the material quickly and will allow him to retain the material better. Now, while some of us learned SAS exclusively from the User's Guide, many first-time SAS users are not so fortunate. Prior to the publication of the Applications Guide many users failed to appreciate the full power of some SAS statements. How many SAS beginners, for example, can tell you after reading the User's Guide that

- a MACRO can be used for other than simple string substitution
- a colon has two distinct uses (he cannot find any reference to the colon in the Index)
- an array can be character as well as numeric?

This is not meant to disparage the User's Guide in any way. It is, clearly, meant as a reference manual; that is why SAS Institute publishes both the Introductory Guide and the Applications Guide. The point here is that in presenting technical information to a learner, conventional way of writing has its problems. Consider for a moment the fact that while the amount of technical information with which we have to cope increases exponentially from year to year our manner of communicating these information via the printed page really has remained unchanged.

The trouble with conventional prose is that paragraphs and unnecessary transitional phrases get in the way of essential ideas. There is a good lesson to be learned from speed-reading courses. Here, we are taught not to read the way we write - not to read sentences word by word - but rather to scan paragraphs for main ideas. But, if we are supposed to scan paragraphs only to hunt for main ideas, why should we write paragraphs in the first place? What we need is a paragraph-less way of writing that readily communicates information to those who are learning it for the first time as well as to those who are trying to recall previously-learned materials.

That, in essence, is information-mapping.

I have used information-mapping in my Chemistry courses at the University of the District of Columbia since 1978 (Olympia, 1979). That experience taught us that students quickly learn and retrieve materials from information-mapped manuals. We observed, for example, that while most of the students' textbooks were heavily marked with highlighting pens very few of the Information-mapped materials had any such markings.

This paper describes the use of information-mapping in SAS training courses. In particular, it describes how the use of an information-mapped manual can improve a student's ability to learn and retain materials better both for initial learning and reference learning.

WHAT IS INFORMATION MAPPING?

Horn (1975) defines information-mapping as a "method of bringing together current research and instructional technology into a comprehensive materials development to improve (the delivery of) technical information". One design principle upon which IM is based is "cuing". In an information-mapped manual the student is always prepared for the learning that is about to take place because a material (i.e., a map) whose purpose is to explain a concept does not look quite the same as one that describes how to do something, or one that explains how a process takes place. Figure 1 shows an example of a "concept" map, that is, a map designed to explain a concept - in this case, the SAS Macro. Note that a map contains a title and a collection of Information Blocks with labels on the left margin and separated from other blocks by horizontal lines. Our SAS training manual, then, is nothing more but a collection of related information maps clustered into course units; the maps themselves are built out of well-defined and functionally distinct blocks. Figure 2 is a schematic of the manual organization.

The information map in Figure 1 has several advantages over conventional
prose that teaches one about SAS macros:
- Each information blocK delivers specific information; a definition block defines the term and does nothing else; an example block gives and explains an example and does nothing else.
- The student does not need to wade through unnecessary and complicated transitional phrases.
- The modular form readily permits the revision of materials.

There are roughly 49 commonly used blocks and six basic types of maps:

<table>
<thead>
<tr>
<th>THIS MAP</th>
<th>ANSWERS THIS QUESTION</th>
<th>====================</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept</td>
<td>What is it?</td>
<td>---</td>
</tr>
<tr>
<td>Procedure</td>
<td>How do I do it?</td>
<td>---</td>
</tr>
<tr>
<td>Classification</td>
<td>What kinds/types?</td>
<td>---</td>
</tr>
<tr>
<td>Process</td>
<td>How does it work?</td>
<td>---</td>
</tr>
<tr>
<td>Fact</td>
<td>What are the specs?</td>
<td>---</td>
</tr>
<tr>
<td>Structure</td>
<td>How does it look like?</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>What are its parts?</td>
<td></td>
</tr>
</tbody>
</table>

To explain to someone the steps he needs to follow to run SAS under TSO requires a Procedure map; to explain to him how the SAS supervisor works requires a Process map.

Except for all-purpose blocks such as Introduction and Comment, an information block is typically found in one and only one type of map. For example, a Procedure Table block or a Decision Table block appears in a Procedure map while a PERT Chart block appears in a Process map.

All maps of a given type have a consistent format. While a student who reads a paragraph seldom has any advanced warning of the type of learning that is about to take place - a new term will be defined, a procedure will be explained, etc. - that student faced with, say, a concept map knows that some new term will be defined, rules of usage set, and so on.

OTHER EXAMPLES OF SAS INFORMATION MAPS

This section shows some of the maps in our SAS training manual.

Figure 3 is an example of a Procedure map which already should be obvious from the title. "How to ..." tells the student how to accomplish a commonly desired objective - that of passing a variable value to a TITLE statement during execution time. It illustrates the use of two "concepts", the SYSPARM option and the %INCLUDE statement without explaining what they are. We explain what they are in two concept maps. In this map we just tell the student what he is to do to get the job done.

Figure 4 is a Concept map with a twist because while it addresses the same "concept", i.e., the colon in SAS statements, it really describes the dual use of that one concept. I wrote this after my repeated inability to find any reference to the colon in the Index of the User's Guide. Note that the Reference block points the student to the appropriate pages in the Guide.

Figure 5 is an example of a Classification map. The title, containing the phrase "Types of ..." tells us that. The Classification List block can be replaced with the more graphic Classification Tree block, e.g.,

<table>
<thead>
<tr>
<th>SAS variables may be classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Figure 6 is a map explaining the PATTERN statement of SAS/GRAPH. I use these information blocks to explain each SAS or SAS/GRAPH procedure:
- Use - what it does; when to use it
- Limitation - if any
- Syntax
- Options
- Examples - loads of it.

INFORMATION MAPS IN SAS TRAINING

At a Federal agency where I serve as a consultant, I conduct a bimonthly training course on Basic SAS and SAS/GRAPH. The class is usually full with its limit of 25 students (who have extremely diverse backgrounds and needs). The students are required to
- have experience with TSO
- have a work-related DP problem
- read a locally-written SAS Primer at least one week before class.

The third prerequisite is important because the course is quite intense, meeting only on two successive half-days. The students are required to turn in the results of a run of a non-trivial SAS problem assignment before they can be awarded their training certificate. They have 40 days from the end of the class to complete their assignment. The students are free to call the telephone hotline (which we maintain for all users of the Agency's computer center) for assistance. The hotline calls convinced us early on that the locally-written Primer which
teaches SAS by examples, was insufficient. We also found that many of the students had difficulty with the User's Guide (A copy of the Guide was available in each library and Terminal Room). The SAS/GRTAPH segment of the course was a recent addition to the course. The handouts for this segment are exclusively information-mapped. We found that the students have better success with that. We just recently rewrote the Primer using information mapping. That required the use of supplementary maps such as Overview, Summary and Table of Contents maps.

The first real test of the new Primer came when 10 administrators at the University of D.C. requested a SAS course to enable them to obtain reports and summary statistics from the student databases. None of the administrators had any data processing experience whatever so the first day of the 4-day course (spread over 4 weeks) concentrated on the usual Introduction-to-Computers materials. I believe that the use of the information-mapped manual helped make this class a success. The less strenuous class schedule clearly helped also.

CONCLUSION

In SAS training courses, as in most other courses, a student really begins to learn the material when he is faced with an application problem that needs solving, not while he is listening to the lecture. If, for one reason or another, his lecture notes are insufficient to help him solve his application problem, he needs a handout or manual that will allow him to locate the material quickly and will allow him to retain the material better. I believe that information mapping excels in these two areas.

REFERENCES

Information Mapping is a trademark of Information Resources, Inc.

THE GAS MACRO

Define a GAS macro, a segment of a SAS statement, or a set of SAS statements that becomes inserted in the program that refers to it by name.

Example 1

$MACRO define February 14, 1986
A statement that refers to this macro, such as:
PRINT REPORT AS OF February 14, 1986.

Example 2

$MACRO select
IF SEX = 'M' THEN SELECT 1;
Inserting the word 'select' in the DATA step of a SAS program will cause all observations with SEX='M' to be selected.

Example 3

$MACRO variable
INPUT VARY $ SEX $ GRADE $;
Records of data to be processed.

Rules

a macro must begin with the word MACRO, immediately followed by the name which should be a valid SAS (variable) name.
a macro must end with the $ sign; if the macro itself contains the $ sign it should be coded as two successive $.
a macro must be defined before it is used.
a macro may invoke other macros, up to a nesting level that is 8 deep. A program may use up to 64 macros.
a macro can be used in a separate library instead of being in-stream; in this case, the macro library must be concatenated in front of the program that uses it during SAS invocation.

Use

a macro to
a) eliminate coding of segments that are used repeatedly in a program, e.g., a variable list.
b) facilitate structured programming; a complete program can be built out of a series of macros.
c) allow different SAS programs to share common segments stored in a macro library.

FIGURE 1. An example of a concept map

FIGURE 2. SAS manual organization
HOW TO PASS A VARIABLE TO A TITLE STATEMENT DURING RUN TIME

Introduction
Suppose you have a SAS program that produces a customized report whose title goes something like this:

\texttt{TRANSACTION REPORT AS OF repdate}

where 'repdate' has a value that changes from run to run.

In the past, this problem is solved by defining repdate as a macro and changing its value each time a run is made.

Using the SYSPARM option and the \%INCLUDE system macro it is now possible to supply a parameter to your SAS job (as in a title statement) during run time without changing your code from run to run.

Use
Use this procedure to pass a value to your title statement during execution time for a job running under TSO.

Comment
This procedure can be extended easily to pass any value for use by your job in statements other than TITLE.

Procedure Table

\begin{tabular}{|c|c|c|}
\hline
STEP & PROCEDURE & REMARK \\
\hline
1 & Allocate a small data set to a ddbname say, TITDD, that will receive the Title stmt & \\
\hline
2 & Precede your program with: OPTIONS INCLUDE;
DATA NULL; FILE TITDD;
\texttt{X=SYSPARM();} PUT 'TITLE TRANSACTION REPORT AS OF ' \texttt{X}';
RUN;
\%INCLUDE TITDD; & This is a one-time change; the set of statements may be assigned to a macro also \\
\hline
3 & Allocate your program to SYSIN I/O files that your pgm will use & Allocate any other I
\hline
4 & Issue this TSO command:
\texttt{sas options('sysin=sysin sysparm='"""string"""')} & \texttt{<string>} is the parameter you want passed to the title. Note the 4 and 5 single quotes around the string. \\
\hline
\end{tabular}

FIGURE 3. An example of a procedure map
THE COLON IN SAS STATEMENTS

Introduction
In SAS the colon (:) has two different meanings depending on the context in which it is used.

Use
Use the colon
- in an INPUT statement to specify a format to a variable that is being read by list input mode (USAGE 1)
- in a statement that compares two strings of different lengths so that the longer of the two is truncated to the length of the shorter one before the comparison is made (USAGE 2).

Example
USAGE 1:
INPUT EMPNAME : $25.;
EMPNAME becomes implicitly defined with length 25 instead of the default 8 characters with conventional list input.

USAGE 2:
IF EMPNAME = : 'SMITH' THEN DELETE;
This deletes all observations whose EMPNAME variable begins with SMITH. Without the colon, and assuming that EMPNAME has a length of 25, SMITH (the shorter operand) will be extended with 20 blanks before the comparison is made, causing a mismatch condition in practically all cases.

Rules
USAGE 1:
- The colon must appear between the variable name and its format, separated from either by at least 1 blank.
USAGE 2:
- The colon must appear immediately after the operation symbol.

References

FIGURE 4. Dual concepts in a concept map

TYPES OF SAS VARIABLES

Introduction
SAS variables are often defined for the first time in the INPUT statement. In this statement what follows the variable name, e.g.,
- a dollar sign
- a specific format
- a range of column positions and nothing else tells SAS what kind of a variable it is.

Classification
Sas variables may be classified as

List
- Numeric
  - standard integer
  - standard non-integer
  - date
  - time
- Character

Related pages
INPUT statement, 4
SAS Formats, 16

FIGURE 5. An example of a classification map
The PATTERN Statement

**Use**
Defines colors and patterns (shading and hatching styles) to fill areas within a SAS/GRAPH picture.

**Comment**
For example, various bars of a bar chart produced by PROC GCHART may be drawn with different colors and crosshatching lines. The pattern may be selected from a predefined set or may be user-defined.

**Placement**
Global (anywhere in the program)

**Limitation**
Under Release 79.5, PATTERN affects the output of only these PROCs: GCHART, GCONTOUR, GMAP and GPLOT.

**Syntax**
PATTERN(n) C=color V=value;
Omitting both C- and V- options cancels the pattern set by an earlier PATTERNn statement.

'color' is any valid color recognized by SAS/GRAPH, or its abbreviation.

'value' may be any one of the following:
- X1-X5 --> crosshatched patterns with predefined color & density
- L1-L5 --> negative slope parallel lines
- R1-R5 --> positive slope parallel lines
- S ------> solid color
- E ------> empty (no pattern)
- abxxx --> user-defined patterns where
  a = 1-5, specifies line density (5 is most dense)
  b = X, for cross-hatching
  N, for NO crosshatching
  omitted, for parallel lines
  xxx = 0-3X0 or omitted
  specifies the angle of the lines drawn; if omitted, parallel lines are drawn vertically.

**Usage**
- In PROC GCHART, PATTERN1 is used to fill the first area defined in the SUBGROUP= specification; PATTERN2 is used to fill the 2nd area, etc. If the PROC GCHART statement does not include the SUBGROUP= specification only PATTERN1 is used.
- In PROC GCONTOUR, when the PLOT statement contains the PATTERN option, PATTERNn is used to draw contour level n.
- In PROC GMAP, PATTERNn is used to draw level n of the response variable in a choropleth map.
- In PROC GPLOT, when the PLOT statement contains the AREAS= specification, PATTERNn is used to fill area under the curve starting from the horizontal axis. Only valid values of the V= option of the PATTERN statement are S, E, abxxx.

**Examples**
PATTERN1 C=RED V=S;
PATTERN2 V=X1X345;
PATTERN3; (cancels previously set third pattern).

**Figure**
6. A SAS/GRAPH information map