SUSIe - A TEXT FORMATTER FOR MVS

Klaus Daube
Oerlikon Bührle Rechenzentrum AG, Zürich, Switzerland (S216)

Abstract

SUSI is a text formatter for the MVS operating system with TSO/ISPF on IBM mainframes. Most commands describe textual entities and only few commands describe typographical aspects. The presentation of a document depends on the output device and a profile unique to a type of document. A key feature of *SUSI* is support of structures above the "document level" to build books and parts from documents.

A system of panels (menues) guides the user through all available tasks. Besides editing and browsing of texts they can be formatted both in foreground (on the screen) or in background by means of generated job control. A complete tutorial on all *SUSI* commands as well as text skeletons for editing the text are available.

Introduction

A company like ours with IBM mainframes for "common EDP" doesn't have many choices for formatters, input and output capabilies. So besides the opportunity to get deeper knowledge and competence in document processing the development of *SUSI* aimed at:

- using existing resources (TSO, ISPF, alphanumeric screens with common EDP keyboards, line printers etc.).
- processing lenghty documents rather than letters, mailing lists etc.
- allowing different character sets, styles and emphasis for the representation of a document.
- the layout of documents must be standardizable, since more than one author may work on one document.
- enforcing some kind of structure to improve readabiltiy of documents
- supporting the organisation of manuals where we have BOOKS, consisting of PARTS, which are divided into documents.
- avoid typographical cruelties like indentations over half a page or substructuring ten levels deep with mixing of alphabetical, roman and arabic numbering.
- *SUSI* should be usable by people familiar to VDU's.

At the beginning of our programming we thougt to finish the project within 8 man-months. At that time we didn't know that the

developement of a document processing software may be an endless loop of user experience, ideas for enhancements and the integration of new features. .

ISPF (Interactive System Productivity Facility) with PDF (Program Developement Facility) offer the tools or even machine tools to build a productive environment for the user. It was a bootstrapping process to build the current system "*SUSI*". The formatter and its use asked for support and guidance, ISPF offered new capabilities and so on. Today the non-sophisticated user often does not distinguisch between *SUSI* and "the rest".

Support of "superstructures"

SUSI is designed to support libraries of manuals. For that purpose a hierarchy of three levels is specified: The library spans all the information concerning a BOOK. BOOKS are divided into PARTS which are divided in documents. In this case documents are called CHAPTERS to distinguish them from common documents.

This hierarchy is reflected in the numbering scheme of the CHAPTERS. The text you are reading is a chapter within the book 410 called "Dokumentverarbeitung am MVS System". The current PART has the number 10 with the name "Allgemeines zur Dokument-verarbeitung" And this is chapter 8. So the complete number of the chapter is "410.10.08".

Since there is no hierachical file system in MVS it is not very easy to reflect the logical structure of a document with a physical structure. We thus implemented a numbering scheme for the members (files within the library of text). The library itsself is named after the BOOK (e.g. S.HB410.TEXT). Within the library the highest rank is PART, in this case named H10. The chapter is named H1010. Pieces of the chapter (for imbedding) are named H1010#xx.

The name of the BOOK and related definitions like author, type of pagination etc. is specified in a library member with the name H. The name of the PART and related definitions is specified in the member with name H10 (for PART 10). When processing a chapter the defintions for BOOK and PART are taken from these members. So the author of a chapter does not neeed to know this information. These members are normally maintained by a text administrator.

These facts can be illustrated by the picture on the next page.

SUSI makes it possible to produce a table of contents or an index not only of a single document but also of a part or a book. Both for table of contents and for index separate libraries are set up. Their file names are suffixed by CONTENT and INDEX, respectively. These libraries are not touched by the *SUSI* user directly. Their structure is similar to the text library (but without the members for imbedded text). For manual libraries automatic storage of the information for table of contents or index is specified. But also temporary production is possible.

It can happen that text members are modified after the corresponding information for index or table of contents were stored. When formatting an index or table of contents, *SUSI* checks for such a condition, and issues appropriate warnings.

```
+---- S.HB410.TEXT ------------------------------------------------+
|                                                                   |
|   +----- H ---------------------------------------------------+   |
|   |                                                           |   |
|   | .ASSIGN BOOKTIT, Dokumentverarbeitung am MVS System       |   |
|   | .SET    BEREICH, DTA                                      |   |
|   | .ASSIGN PAGINATION, CHAPTER                               |   |
|   +-----------------------------------------------------------+   |
|                                                                   |
|      +----- H10 ----------------------------------------------+   |
|      |                                                        |   |
|      | .ASSIGN PARTTIT, Allgemeines zur Dokumentverarbeitung  |   |
|      | .SET    DV, Dokumentverarbeitung    ;.* short hand     |   |
|      | .SET    TV, Textverarbeitung        ;.* short hand     |   |
|      +--------------------------------------------------------+   |
|                                                                   |
|         +----- H1010 ------------------------------------------+  |
|         | .CHAPT 410.10.10, SUSIe- A TEXT FORMATTER FOR MVS   |  |
|         | .*-------------- text for SEAS AM 85 in Zuerich, sep 85|  |
|         | .SET S,      $9$SUSI$-9$                            |  |
|         | .SET cent,   $23$$$$-23$                            |  |
|         | .SET DP, do|cu|ment pro|ces|sing                    |  |
|         | .*                                                  |  |
|         | &TITLE&                                             |  |
|         | .space 2                                            |  |
|         | &AUTHOR&                                            |  |
|         | &COMPAG&                                            |  |
|         | .........©....etc. .............                    |  |
|         +-----------------------------------------------------+   |
|                                                                   |
+-------------------------------------------------------------------+
```

Formatting

In general *SUSI* handles four types of documents. They are introduced with the appropriate commands:

DOCTITLE defines a general document which is normally not embedded in a hierarchical environment.

CHAPTER defines a document which is normally embedded in a hierarchical environment. "Above" the chapter a PART of a book is defined and above this a BOOK is defined. The titles both of BOOK and PART are implicitly defined by the number of the chapter. Also the names of the libraries containing entries to tables of content and indices are implicitly defined by it.

CONTENT defines the document "table of contents". This document may be introduced by an arbitrary text (e.g. explanations). It is formatted as specified by the profile. The contents of this document may come from a temporary file being the output of the last formatting process or from a library.

INDEX defines the document "index". This type also may be be introduced by an arbitrary text (e.g. explanations for the various notations used). It is formatted as specified by the profile. Currently this is the only document type which can be formatted in multiple columns. The contents of this document may come from a temporary file being the output of the last formatting process or from a library.

Specification of layout

The layout of a document is specified by a "profile". Chosen system and user variables are set depending on the desired output device. These specifications concern

- heights of the page, length of line, position of first print line
- headers and footers
- form of revision mark
- specification of special graphics like national use characters, bullets etc.
- representation of section headings
- representation of index and table of contents
- highlighting of catchwords in the text
- general format and hyphenation
- "company standards" (see text skeletons)

These profiles force a more or less user independent representation of the documents. Someone has to account for these definitions because for some people this approach is very hard to understand. Everybody being able to write imagines himself an expert in typography...

If one only wants a temporary table of contents or set special variables only for that run, he can do that by entering runtime parameters. They are written with normal commands. Of course there is a special panel defined for this purpose.

Table of contents and index

Both table of contents (TOC) and index can be formatted very flexibly. The TOC can be "resolved" to a certain level of detail. For a TOC of a chapter or document the whole structure of mainsections down to the subsections is needed. For a TOC for a BOOK only the entries of PARTS, chapters and mainsections may be needed. This can be specified at the formatting of the table of contents.

The index gathers main information from short titles associated with the book, part, chapter and section headings. Further informations stems from the catchwords in the text. There are also commands for direct entry of catchwords, synonyms and references to other catchwords. Even having these instruments it is not easy to produce a meaningful index. We still have to learn the rules of index making.

When formatting the index the user can choose if he wants a full blown index with both forward and backward refrences or a more concise form. The representation of the varios entry types and other formatting rules are defined in the profile using preset variables.

Both TOC and index are documents of their own kind. Hence they always start on a new page. Their main contents can be preceeded by arbitrary text like any document or chapter. Thus it is easy to place introductary text or explanations.

Imbedding subsidiary text

Text may be imbedded either by COPY or by INCLUDE. The first works in a very common way. The input is "switched" to another file. INCLUDE supports the structure of a text.

When a main section includes a file (or part of it), this file only may start with arbitrary text on the main section "level", if there is no such text in the "calling" level. Included text is some kind of a local environment. Character sets, formatting and constructs like lists are "closed" at the end of the imbedded text. Also variables specified in the imbedded text only have local scope. These mechanisms support proper structuring. They are needed to use text not only in one place. One rule of our handbook is redundancy. So some pieces of text are used in several places.

For copied text there are no such "restrictions". Even incomplete commands can be imbedded. This construct is used for supporting text skeletons or "canned text". Since a *SUSI* variable may contain commands, the function of macros can be simulated very closely.

Both commands can specify the imbedding of a part of a file. This part may be defined by:

- First and last line to be imbedded. Both may be defined either by number or by a string match.
- The name of a programming language. In this case blocks of comments are extracted from the program source.

Special tasks

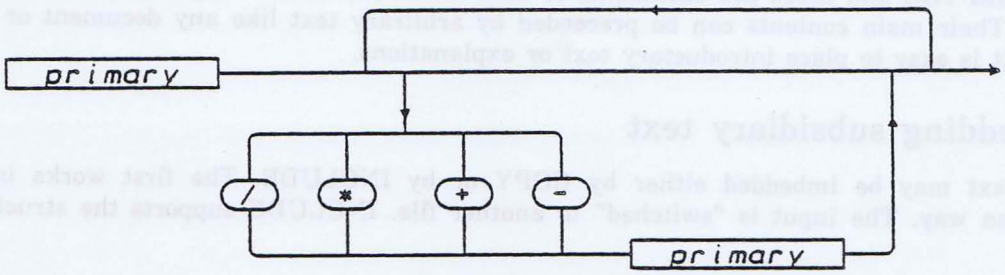
Hyphenation currently uses an algorithm developed at the ETH Zürich, which uses a small dictionary of prefixes, infixes and suffixes as well as indivisable sequences of consonants and a small list of real exceptions. This works "sufficiently" well for German. This means that *SUSI* hyphenates incorrectly less often than the "common user".

For US-english we tried to define Knuth's first algorithm from TEX. It is very lazy in finding hyphenation points. So we did not implement it.

SUSI is intended primarily to format manuals. They are changed and updated very often. Marking of revisions can be done automatically by using the flags produced by the ISPF editor. They can be reset by a function of ISPF.

Since *SUSI* is "still under construction", occuring errors can be reported. If this task is choosen from the main menu, the current text and related information - like output device or profile in use - are copied to a special library. The user is then prompted to enter information about the error. He also may enter information about a circumvention found. The developement team is automatically alerted by a message.

SUSI does not have any macro facility. For special purpose we use the general macro processor ML/1. Currently this is only possible in the manner of preprocessing the pieces of text. Most use of this tool is made for generating syntax diagrams from an input manageable on ordinary screens. Currently only printer VARIO has all the graphics available.



Output from VARIO printer

Multiple column printing is only available for indexes now. We didn't make it generally available because on EDP printers the characters are to large and the lines to short. We plan to support multiple column printing on page printers.

Supported printers

ASA common EDP printers with fixed pitch, only one character set, overprinting and underlining.

TSOSCREEN for foreground presentation on the screen.

VERSATEC an electrostatic printer with only one character set, no overprinting, no underlining

VARIO a driver for the VERSATEC printer plotter with 5 sets of characters in three widths (half, normal, double) and three heights (half, normal, double). Characters can be slanted, underlined, bolded and shown in revers. It is very easy to specify arbitrary shapes of characters by the user. Also super- and subscripting, absolute and relative placement on a line are possible. The character sets defined for VARIO are the reference to define special graphics (like $21$6-21 for ¶).

XEROX uses the definitions for the Xerox 2700 page printer. *SUSI* supports 3 sizes of characters, normal and italics, super- and subscripts, bold and underline.

The table of the reference character sets for *SUSI* shows five sets with 94 graphics each. The first is identical to US-ASCII. The first and second are identical to ISO 6937 or VIDEOTEX. A special feature of the second set are the "flying accents". The third set uses HP's mathematical set from the 264x terminals as a base. The fourth holds the special characters from DEC's VTxxx terminals. The last holds miscelleneous semigraphics. On printers other than VARIO the actual graphic substitute depends on the fonts available.

Commands

Many of the commands communicate with "preset variables". For example the command CONTENT uses (sets or reads) the variables CONTFORM, CONTBOOK, CONTPART ... CONTSSEC, CHAPTTIT, CHAPTNUM and PAGE. There are about 50 preset variables supporting the commands, layout, writing of date etc.

The names of the commands of *SUSI* are in "English", since German names often are very long and will not be unique in the first few characters. So they could not be abbreviated in a simple manner.

The commands are started at the beginning of an input line with a period (similar to SCRIPT, giving advantages in ISPF). Both commands and quoted strings may be continued with a + at succeeding lines. More than one command can be placed on a line, if the command is terminated by a semicolon and the command introducer "." follows immediately.

Parameters for commands are either keywords like KEEP, NONUM, ARABIC (which may be abbreviated to 3 characters), numeric values (which may be dimensioned) like 3p, 5mm, 170dp, or strings. A string may be written as normal text, if there are no commas or semicolons in it and no continuation on the next line is necessary. In the latter case the quoted form is to be used. A special form of string is the qualified string like `X'27ad' T'100;200 B'` allowing to include hexadecimal data in strings (e.g. to write escape sequences). Both the names of commands and of key words are not case sensitive.

Table of Commands

command	usage
ASSIGN	assign "preset variables": - date, time, page number etc. - parts of filenames - titles and definitions for BOOK and PART - layout variables, definition of printer - representation of headings - representation of revision marks - representation of index and table of contents
BOUNDS, EBOUNDS	set text boundaries.
CHAPTER	define a chapter (document within a manual library).
COMMENT, *	comment for the human reader.
CONTENT	produce table of contents.
COPY	imbed secondary files or parts thereof (may be nested).
CWENTRY	specify catchword which is not present in text.
DOCTITLE	define an arbitrary document.
FOOTNOTE	place footnote.
FORMAT, EFORMAT	define how input lines have to be transformed to output lines (wrapping of words, handling of blanks, treatment of special characters).
HYPHENATE	chose hyphenation (language or on/off).
INCLUDE	include secondary file or part of it considering structure elements.
INDEX	produce index from catchwords, synonyms, references and short titles.
KEEP, EKEEP	keep parts of text together.
LIST, ELIST	numbered, unnumbered and definition lists.
MSECTION, SECTION, SSECTION	define text units.
PATTERN	specify layout lines (headers and footers).
REFERENCE	specify references to a catchword
SCOPE, ESCOPE	specify scope of text (output device, audience)
SET	assign user defined variables.
SPACE	space, jump to next page, space for picture with text.
SYNONYM	specify synonyms.
TADEF, TAHEAD, TATEXT, ETABLE	define table header, columns and their formatting, separation of table elements. Table heads are repeated on each page.

Lists

Since lists are very often used in texts (at least in our texts), this command may be shown by a simple example:

```
.list def, keep, 20mm, 2mm, $4$
'ASA' used for identifying asa printers in profiles
'VER' identifies the VERSATEC printer in the
      OBRZ profiles
'VAR' identifies the VARIO software printer
      in our profiles
.elist
```

This will produce

ASA	used for identifying asa printers in profiles
VER	identifies the VERSATEC printer in the OBRZ profiles
VAR	identifies the VARIO software printer in our profiles

Tables

Tables are defined by a set of commands. TADEF specifies the characters used for tabulation and "new line" as well as the placement within the page (left, right, center). This command is followed by a line defining the columns with a raster. Columns may have fixed width (defined by a distinct amount of blanks) or may be adjusted to the space available (width defined by a "blank filler". The lines after this raster define a "separator", which can be placed at any line with a simple reference.

The heading of a table is introduced by the command TAHEAD, which specifies the formatting of the columns (left, right, center). There is no block formatting within tables.

The text of the table is introduced by TATEXT. If the table should act as a place holder on the page, a parameter FILL specifies filling of the rest of the page with the raster only. Also the amount of space between "table elements" can be specified. If the elements are to be kept together (forming a paragraph), the parameter KEEP can be set.

SUSI cannot nest tables within lists or vice versa. Also tables cannot be nested. But a table may use a TAHEAD or TATEXT only.

An example may illustrate this construct. Not all features are described above.

```
.TADEF >, <
$23$x          x% %_x_%  %x
$23$tqqqqqqqqqn%q%qnq%qq%u
.TAHEAD CENTER, RIGHT, LEFT
- number > what to do > what you will get
--
.TATEXT 1L, KEEP
- abc    > alphanumeric input > somthing very similar
- ABC    > same as above      > all the stuff in upper case
- 123    > count on your fingers
+        > wrong result, if counted wrong
--
- etc    > and so on  > per pertuum mobile
.ETAB
```

The raster specifies a table with 3 columns:

- the first column is 10 places wide
- the second column 1 unit wide
- the third column is 2 units wide

One unit is determined by the space available after subtracting the width of the fix columns from the line length. This table will look like the following. The table uses the full width of the line. Of course that could be changed.

number	what you may see	what you will get
abc	alphanumeric input	somthing very similar
ABC	same as above	all the stuff in upper case
123	count on your fingers	wrong result, if counted wrong
etc	and so on	per pertuum mobile

Character Sets and Output Control

The output control is based on a virtual printer. Most of its properties are available on the real printer VARIO. The input to *SUSI* only uses characters from the ASCII set. The various pi-characters needed are supported by several reference chararacter sets. On a specific printer *SUSI* produces the best possible result (e.g. display an * instead of a bullet on the TSO screen).

Within the input text some special characters are used to control the formatting process as well as the final output. All of these are used in pairs and may be nested like $4,&high1&$. If one of these special characters is needed as a graphic itself it is written double like $$ which will produce a $ sign on output.

$ specify alternate character set and highlighting as well as different styles of the character sets like $4,5,9$ for "bold, double high, italic".

¢ specify catchwords like ¢human brain¢ or ¢mistake 17¢. The non ASCII graphic ¢ is used in our environment, since it is translated to ^. So ¢ and ^ are equivalent graphics in OBRZ (not for *SUSI*).

_ defines a pseudo blank or glue to keep characters or words together.

& use of a variable like &DATE& giving 23.9.1985. Names of variables are case sensitive. Also most special characters may be used in variable names like &=>& (if defined by a SET command)

% defines fillers which are expanded prior to the formatting of an output line (e.g. %. .% gives . . . .).

¬ defines control sequences which are directly transferred to the output. This may be used for special features of output devices not known to *SUSI* like forms overlay or imbedding of raster images.

| the vertical bar defines hyphenation points in a word. No automatic hyphenation is done for a word containing |. This implies that a word preceeded by this graphic will not be hyphenated.

Of course it is possible to use national keyboards with *SUSI*. But these texts will not be readable on other VDU's not supporting national characters like ä or Ü. It is also necessary to have unique code points defined for all the national use graphics for one installation. Otherwise the meaning of a code can not be determined. *SUSI* supports these special codes with a translation of one character to a variable.

System environment

Although a text formatter is usually running in batch mode, *SUSI* is imbedded in the dialog system TSO/ISPF of the MVS operating system.

The program itself is written in PASCAL using some 20'000 lines of code, containing about 550 procedures. For special purposes (allocation of files, obtaining properties of files, reading directories etc.) assembler routines are called. The program code needs about 400KB main storage. A reasonable data memory size is 200KB or more. Thus running *SUSI* under ISPF a region size of 2MB is recommended because ISPF uses about 1MB.

Before text can be formatted it has to be created and manipulated in various manners. These tasks are common to all types of text, like documents, programs and data. ISPF with PDF is a dialog handler, featuring panels (screen layout), programs and command lists (for action), help and tutorial texts. In response to incorrect input a short message is presented. By the push of a button (program function key) you can get a context related short explanation. By the next push you get a whole tutorial panel. A feature called "file tailoring" allows easy generation of job control for batch runs or of data files. If a user doesn't know what to enter into a field of the panel the help facility presents him possible entries. Since panels are interpreted, the behaviour of the system can be changed (or corrected) very quickly.

Starting at a main menu a piece of text can be

- edited
- browsed
- formatted and presented at the screen
- formatted and printed in the background
- encrypted and decrypted by a simple algorithm

All tasks around *SUSI* can be performed in a "split screen" mode, doing one task (e.g. looking at the error messages from a formatting process) in one part of the screen, and another task (e.g. editing the text) in another part. All of the tasks also can also be called from a panel showing the directory of a text library. The details or pieces of this "system" shall be described now.

All text which we have made ourselves is in German. To avoid strange translations of technical terms we left them in the original wording. So you see words like "foreground" or "batch" on our menus. We know that this may be not really "user friendly", but the German counterparts often are both imprecise and to long for a menue.

Main menu

The first line of the panel shows "where you are": in the OBRZ text processing environment. Generally the user can enter data into fields preceeded by ===>. To select an action the field OPTION is specified. This field also serves for special actions like leaving the dialog system, invoking programms directly and so on.

The list of options (what can be done further) use terms which are used in ISPF and PDF for functions already present there (E, B).

Names of libraries for text and data are composed of three parts: the project, the library and the type. A member of the library is a piece of text. This may be a complete document or only a part of it intended for imbedding anywhere. The fileTYPE is associated with information for the editor. If one don't know which member of the library he wants to process, a blank field MEMBER leads to a list of the members (directory of the library). Since not all datasets (files) in an installation conform to the three level scheme, arbitrary names can be entered into ANDERER SEQUENTIELLER ODER PARTITIONED DATASET.

With the PROFIL NAME the type of the document is specified for formatting. These descriptions of the presentation are held in a library too. But they also may be in a file of the user (what is not recommended).

For formatting some RUNTIME PARAMETERS may be set. These override the definitions in the profile and are of temporary scope.

In our installation the printers don't have a unique name. So the number of definitions necessary vary. At most the user must specify the *SUSI* name of the printer (AUSGABE-GERAET), the DESTINATION (an address within the network) and the output class AUSGABE-KLASSE. An underlying table guides the user, if he enters incorrect values.

```
------------------------ OBRZ / TEXTVERARBEITUNG ---------------------
OPTION  ===> _

E Edit      F SUSI Foreground    M SUSI Meldungen    R Fehler Registr.
B Browse    J SUSI Batch         I SUSI Info         Q Verschluesseln
                                                     leer Member Liste
ISPF LIBRARY:
   PROJECT ===> S
   LIBRARY ===> HB410
   TYPE    ===> TEXT
   MEMBER  ===>                (ausser fuer eine Member Liste)

ANDERER SEQUENTIELLER ODER PARTITIONED DATASET:
   DATASET NAME ===>

PROFIL  NAME     ===> RHB
   oder DATASET ===>

RUNTIME-PARAMETER ===> JA     (JA oder NEIN)
          DATASET ===>

AUSGABE-GERAET ===> X2700
   DESTINATION ===> SR002          AUSGABE-KLASSE ===> A
```

Member selection list

If no option is specified and the ENTER key is pressed, then a list of the members of the specified library is pesented. Left to the list of the member names the same options as on the main menu can be entered:

```
TEXTVERARBEITUNG - S.HB410.TEXT ------------------------- ROW 33 OF 118
COMMAND ===> _                                        SCROLL ===> CSR
    NAME        VER.MOD  CREATED   LAST MODIFIED   SIZE  INIT   MOD   ID
  FOOT           01.00  84/03/05  84/03/05 15:43     2     2     0  ST#K
  GUGUS          01.02  83/08/12  85/06/24 12:13    40    11    40  ST#K
  H              01.05  82/04/05  84/06/12 17:55     9    18     2  ST#K
  HTITLE         01.01  84/06/12  84/06/12 18:10   106    71    44  ST#K
  H00            01.00  82/08/19  82/08/19 11:59     2     2     0  ST#K
  H0000          01.03  82/08/19  82/08/19 15:01   143   106   120  ST#K
  H05            01.12  84/02/22  84/02/28 13:29     2     2     2  ST#K
  H05CON         01.00  84/03/12  84/03/12 09:37     2     2     0  ST#N
  H0500          01.16  84/02/22  84/03/12 15:18   144    20   134  ST#K
  H0500#11       01.12  84/02/22  84/03/09 13:53   131    69    71  ST#K
*****
```

There is an additional option in this panel to change information related to the formatting process. So the user don't need to leave the member list only to change the profile name etc.

Tutorial

Assume a user not knowing very much about *SUSI*. Selecting `I` on the main menu lets him dive into a rich set of tutorial panels. At first he gets an overview:

```
--------------------------------- SUSI INFO ----------------------------
OPTION  ===> _

  ASS  ASSIGN (Setze res. Variablen)     INC  INCLUDE von Text-Elementen
  BOU  BOUNDS (Rand veraendern)          IND  INDEX (Stichwort-Verzeichn)
  CHA  CHAPTER (Handbuch Kapitel)        KEE  KEEP (Text zusammenhalten)
  COM oder *  COMMENT (Kommentar)        LIS  LIST (Liste, Aufzaehlung)
  COM oder *  COMMENT (Kommentar)        LIS  LIST (Liste, Aufzaehlung)
  CON  CONTENT (Inhaltsverzeichnis)      MSE  MSECTION (Haupt-Abschnitt)
  COP  COPY (Kopieren)                   PAT  PATTERN (Kopf- und Fuss-Z.)
  CWE  CWENTRY (Stichwort erzeugen)      REF  REFERENCE (Schlagwort erz.)
  DOC  DOCUMENT (normaler Text)          SCO  SCOPE (Text Tailoring)
  EBO  EBOUNDS (BOUNDS aufheben)         SEC  SECTION (Abschnitt)
  EFO  EFORMAT (FORMAT aufheben)         SET  SET (Setzen von Variablen)
  EKE  EKEEP (KEEP abschliessen)         SPA  SPACE (Zwischenraum erz.)
  ELI  ELIST (Ende einer LISTe)          SSE  SSECTION (Unter-Abschnitt)
  ESC  ESCOPE (Ende von SCOPE)           SYN  SYNONYM-Eintrag erzeugen
  ETA  ETABLE (Ende einer Tabelle)       TAD  TADEFINITION (Tabelle)
  FOO  FOOTNOTE (Fussnote erzeugen)      TAH  TAHEAD (Tabellen-Kopf einl)
  FOR  FORMAT (Text-Darstellung)         TAT  TATEXT (Tabellen-Text einl)
  HYP  HYPHENATE (Autom. Trennen)

  TABLE Tabellen-Menu                    X Allgemeine Erklaerungen; News
```

Here you can see a complete list of all the *SUSI* commands. Miscellaneous information about how to write the commands, descriptions of "preset" variables etc. are entered by selecting `X` from this menue. Also messages concerning new features of *SUSI* are found here.

The more experienced user can plunge directly to a specific information. To get the news directly he can enter `I.X.NEWS` in the main menu.

This tutorial service uses a special program to extract the information from the same source as the editor models. So these two forms are corresponding all the time and changes and modifications have to be done only once.

The general tutorial service of ISPF normally is entered by error handling services. These texts are left in US-english for the general services like edit or browse. For *SUSI* related (new developed panels) they are in German. For example, if the user enters `Y` in the above information panel instead of `X`, then a short message appears in the upper right corner of the screen:

```
---------------------------- SUSI INFO ---------------- FALSCHE AUSWAHL
OPTION  ===> Y

  ASS  ASSIGN (Setze res. Variablen)  INC  INCLUDE von Text-Elementen
  BOU  BOUNDS (Rand veraendern)       IND  INDEX (Stichwort-Verzeichnis)
  ........
```

Now he can use the HELP facility to get more information on that topic. He enters HELP or presses a designated "program function key" and gets a message of 1 line.

```
---------------------------- SUSI INFO ---------------- WAEHLEN SIE AUS
OPTION  ===> Y
Geben Sie eine der unten aufgefuehrten Moeglichkeiten an.
  ASS  ASSIGN (Setze res. Variablen)  INC  INCLUDE von Text-Elementen
  BOU  BOUNDS (Rand veraendern)       IND  INDEX (Stichwort-Verzeichnis)
  ........
```

With the next HELP-command he enters the tutorial system. In this case he gets the following short description of the *SUSI* tutorial:

```
TUTORIAL ---------------------- SUSI INFO ---------------------- TUTORIAL
OPTION  ===> _

 Fuer alle Anweisungen des Text Formatters SUSI stehen Beschreibungen
 Verfuegung, zur die Sie ueber Option I des Text-Menus anwaehlen
 koennen.  Diese zeigen den Gebrauch der jeweiligen Anweisung; fuer
 genauere Auskuenfte sollte man das Handbuch 415 zu Rate ziehen.

 Der Name der Erklaerung einer Anweisung fuer SUSI wird aus den ersten
 drei Buchstaben des Namens der Anweisung gebildet (man kann, wenn man
 will, auch mehr als drei verwenden).

 Diese Info-Texte sind identisch mit dem Inhalt der MODEL fuer SUSI.

 Die folgenden Informationen muessen explizit mit der entsprechenden
 Nummer verlangt werden :

    1  ASSIGN-Anweisungen
    2  LIST-Anweisungen
    3  Tabellen-Menu
    4  Allgemeine Erklaerungen
    5  SUSI Info Anzeige
```

We hope the user will understand, what we have written. If not, there is also personal service (Help Desk), printed material with examples and also courses are held from time to time. But the "online help" and tutorial is updated most recently und thus up to date.

Editing

It's not the place here to describe the ISPF fullscreen editor in detail. But let's mention some features very useful for document processing. We made *SUSI* use these features:

- Text lines starting with a period, a colon or an ampersand are treated in a special way (this is to support the IBM products SCRIPT and DCF). When ragged text lines are "condensed" with the TF (Text Flow) command, then the rearrangement stops at lines beginning with these characters in the first position. Hence *SUSI* commands start with a period and *SUSI* variables (which may contain commands) are enclosed in ampersands.
- The Text Flow command also stops at empty lines or at lines with a different indentation. *SUSI* has a formatting directive similar to that when specifying `.FORMAT LEFT, WRAP, ASIS` (ASIS leaves all the blanks in the input line).
- To enter text without looking at the screen with his boundaries there is a TE (Text Entry) mode. Only at the bottom of the screen you have to ask the editor for more space. This mode also honours the special characters mentioned above.

- The editior can be called recursively. While editing one text, it is very easy to change some related piece, set up *SUSI* variables in a "heading text" or build an edit macro to support the current work.
- If the lines are numbered (which we recommend), then the last 2 places of the number are used as revision numbers. Each time, that piece of text is edited, this number is incremented.
- The file type (the last qualifier of the name of a file) is associated with properties like recordlength, defaults for screen presentation of the data, numbering of the lines etc.
- Also a class of models is associated with a file type. Models are text skeletons which can be inserted into the current text.
- If the option `RECOVERY ON` is set for a file type then the editor can "redo" a complete editing session that has been terminated abnormally.

Models or skeletons are a very powerful feature of the ISPF editor. We defined models for each command of *SUSI* as well as models containing only explanations. So all the information from the *SUSI* tutorial also is available while editing. The models consist of normal text, which will become part of the edited text, and notes, which will disappear from the text when it is saved.

Let's demonstrate that with an example. The user wants to place the `SYNONYM` command after line 300. He enters an `A` for "after" over the line number and enters `MOD SYN` or `model synonym` into the command line. Then he gets the model inserted from line 310 on. All the lines marked with `=NOTE=` will disappear at the end of the editing session. Only line 310 will become part of the text. Of course it should be adjusted to the users need. The contents of the model is highlighted in the example below.

```
EDIT --- S.HB410.TEXT(H1010#04) - 01.07 --------------- COLUMNS 009 080
COMMAND ===> _                                  SCROLL ===> CSR
****** ***************************** TOP OF DATA **********************
000100 .msect 'Textformatter &S& und umgebung', '&S&'
000200 .************************************
000300 .cwe textformatter &S&
000310 .SYN  'Synonyml', 'Synonym2'
=NOTE=       'Synonyml', 'Synonym2', 'Synonym3', ...
=NOTE=
=NOTE=    Beispiel:
=NOTE=
=NOTE=      .Synonym  Zittergreis, Tapergreis, Tatl, Zausel, Krauterer
=NOTE=
=NOTE=    Beschreibung der notwendigen Angaben:
=NOTE=
=NOTE=      'Synonyml'  und folgende sind Synonyme, die alle im
=NOTE=                  Stichwort-Ver- zeichnis mit ihren Synonymen
=NOTE=                  aufgefuehrt werden sollen.  Mindestens zwei
=NOTE=                  muessen natuerlich angegeben werden.
000400 .syn formatter, textformatter
000500 .*
000600 .sect 'Ziel von &S&', aufgabenstellung
000700 .*================
000800 Es sollen dokumente im vorher skizzierten sinn verarbeitet
000900 werden k&oe&nnen. Der aufwand f&ue&r einfach strukturierte
```

We have combined models with "preset variables" which hide pieces of text and commands from the user. So the models are not overloaded with text. Instead the text is located in members which are copied at runtime (by means of the *SUSI* command COPY). Since with COPY also parts of commands can be imbedded, very powerful constructs can be set up. Also work for which *SUSI* was not intended can be made simplyer with this technique.

- protocols
- memos
- letters
- generating forms (still in use in the era of electronic printing)

Batch processing

The need for job control is one of the major drawbacks on a "productive system". We lost many customers to the world of minicomputers due to this fact. But today this is no more true. For any application ISPF can tailor a job control stream from panels guiding the user. His entries can be checked and choices can be offered. Formatting text is demanding batch processing.

If the user chooses option J (Job Control) for formatting, then most informations needed are already known from the main menu. Information about accounting etc. is held in a skeleton of job control. The user can change this information on demand. The only things left open are "ad hoc" runtime parameters. Most time they are only needed to switch off the automatic marking of revisions. Also the generation of empty pages can be suppressed by this optional menu. But the main use is for specialists only.

If the user do not wish to modify the generated job control (which is normally the case) he is only prompted for some characters identifying his job.

Messages from *SUSI* (error messages etc) are gathered in a special class of output. At our installation we have implemented a system to look at the output on spool as long as it is not yet printed. The message file on spool is only deleted by the request of the user. Optionally he can print it.

Foreground processing

The following picture shows a screen contents from foreground formatting.

```
 SUSI - S.HB410.TEXT(H1010) ---------------------------- Zeile 09 von 66
===> _                                                  SCROLL ===> CSR
------------------------------------------------------------------------

1    Textformatter SUSI und umgebung
     -------------------------------

1.1  Ziel von SUSI
     -------------

     Es sollen dokumente im vorher skizzierten sinn verarbeitet
     werden konnen. Der aufwand fur einfach strukturierte texte
     wie memos oder briefe soll fur einen sachbearbeiter dennoch
     gering sein.

     Das nicht sehr eng definierte ziel fur einen textformatter
     konnte zu beginn der entwicklung von SUSI von keinem be-
     kannten produkt erreicht werden (siehe spatere abschnitte).

         . vorhandener ressourcen (TSO, ISPF, alfanumerische bild-
           schirme, EDV-drucker) sollen benutztbar sein
         . dokumente haben mittleren bis grosseren umfang
         . vielfaltige darstellungsmoglichkeiten sind notwendig
         . ein dokument-typ soll einheitlich darstellbar sein, auch
           wenn mehrere autoren an einem dokument arbeiten
         . mangelnde redaktionserfahrung des sachbearbeiters soll
           durch forcierung bestimmter strukturen ausgeglichen wer-
           den
         . organisationen von handbuchern ist zu unterstutzen
         . typographischen grausamkeiten (mangelnde typographische
           kenntnisse) moglichst verhindern
```

Formatting text in foreground (on the screen) is inteded for previewing and proofreading. If a severe error occurs, the process can be stopped and the errors can be corrected in the text source. All the messages presented on screen are also gathered in a file which can be browsed by option M.

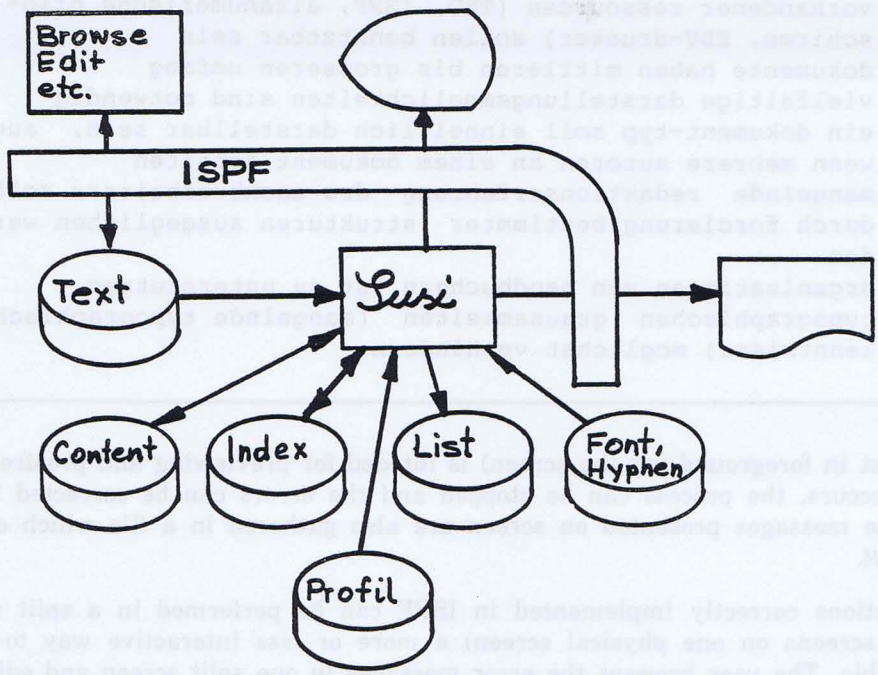
Since all functions correctly implemented in ISPF can be performed in a split screen mode (two logical screens on one physical screen) a more or less interactive way to make corrections is possible. The user browses the error messages in one split screen and edits the source in the other split screen. Of course this is also possible for the error messages gathered on a spool file from a batch formatting.

Since the screens hold 43 lines at most (common are 24 or 32) the screens presented in foreground processing cannot show a total page. Within the formatted page up and down scrolling is possible. A scroll to the right is interpreted as "show next page" and a scroll to the left again shows the bottom of the previous page. Due to storage constraints only the last two pages formatted can be presented.

The screens allow only limited use of emphasis. All highlighting is simulated by "high intensity". For colour screens we may find a better scheme. The screen outputalso is limited to show 80 characters in a line. You cannot see on the screen what you will get on the printer, but you can look at the overall structure, spelling errors etc.

File system

The files associated with *SUSI* are shown in the following picture. The formatted output also can be stored in a library with the same structure as the text library. This may be useful to generate online texts.



As mentioned earlier the lack of a hierarchical file system had to be circumvented by naming conventions for both the libraries and their members. *SUSI* currently can only work on sequential files and libraries (Partitioned Datasets). As long as there is no editor for other file types, this is no disadvantage.

The source lines of the primary text must be numbered. Imbedded texts may consist of unnumbered records. Numbering the lines is recommended for easy reference (e.g. error messages).

The Project *SUSI*

The past

In 1979 when we started the developement of *SUSI* the IBM product DCF (Document Composition Facility) didn't look sufficient for our needs.

- Parameters of macros had to be placed before the macro calls
- Page layout was not alterable after the page was "entered"
- Unfriendly handling of lists
- Almost no handling of tables
- Macros are interpreted. This gives high flexibility but low speed
- Macros produce very often strange error messages

After about 7 man-months the formatter was used for the first time, although most of the more powerfull constructs like LIST and TABLE were not available at that time. The usage by more critical persons (than the programmers of the project) led to useful extensions of the original definitions. Some of these wishes had to be rejected preserving the overriding goal of "forcing some rules of structure".

As noted by various sources it is very hard to have such an instrument accepted by secretaries. One reason may be the necessity of abstraction (you don't see now, what you will get later). But formatting for devices of different level of presentation (and price of final product) calls for a separation of editing, formatting and printing. Today we have text skeletons and good environment. Most people now enjoy using *SUSI*.

My work only was and is specifying the behaviour of *SUSI* and her environment. The programming was done mainly by my collegue Christoph Nievergelt. Andreas Hopplers contributions are in the supporting subroutines, the corrections and modifications of the PASCAL compiler and runtime system. Helping people in solving their documentation problems is our common task.

Further developement

Preprocessing is a good mechanism in a UNIX® environment. In our environment "subtasking" is more adequate. That means calling of subsidiary programs to process a piece of text given to that program by *SUSI*. The result of that subtask is again presented to *SUSI* for further processing. These tasks can be:

- generate semigraphics (boxes, syntax diagrams, wire diagrams, etc.)
- computations (tables, formulas, etc.)
- inclusion of secondary text from various sources like data bases
- obtaining the properties of images and pictures which are imbedded into the datastream by the print server

For many of these tasks a macro processor could be used. To support the subtasking we will widen the file system to have temporary files for multiple purposes like:

- gather information for later formatting (e.g. figure lists)
- share information among the subtasks

A major task will be the enhancement of the hyphenation process. The best solution in our opinion would be one similar to Liang's algorithm used in TEX. But where to get the dictionaries? One also should be able to chose the language of the "built in" texts like the names of the months.

We also should support various character styles (fonts) for the upcoming page printers. For some commands the presentation should be hidden from the user (e.g. FOOTNOTE). And don't forget the personal computers...

This text was formatted by *SUSI* and printed on a Xerox 2700 page printer