EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN – DD /89/ 25

20 JUIN 1989

May 17, 1989

# Text Processing at CERN
# Part I: Overview

Anna Ballanti[1)], Deborah Cork[1,a)], Lex van Dam[1,b)], Jurgen de Jonghe[1)], Eric van Herwijnen[1)], Marco Nijdam[1,c)], Alexandre Samarin[1,d)], and Tony Shave[1)]

To obtain a copy of this document, please use the command

[redacted]ML SGMLPLAN

CM-P00059925

**Abstract**

This report contains a proposal for an integrated text processing environment at CERN on the MacOS, PC DOS and OS/2, VM/CMS, Unix and VMS platforms. Time scales for implementation dates are given in the text. Any system not described in this document will be supported provided it may be integrated easily into the policy that is outlined in this document.

SGML is proposed as the underlying standard that will guarantee both integration through portability at the document level as well as via a common user interface at the input level. Due to the lack of currently available SGML input systems, it will be necessary (in the first instance) to recommend different text processing programs as input systems whilst guaranteeing document interchangeability by providing SGML conversion programs.

When SGML input systems become available we shall migrate to a situation with full SGML implementations at a later stage.

The proposed systems are: MS Word and SoftQuad Author/Editor for MacOS, MS Word for PC DOS, TEX for VAX VMS, IBM DCF and TEX for VM/CMS, TEX for Unix. As desktop publishing system we recommend Interleaf for Unix and VMS based workstations (Apollo/Sun/Vaxstation). This paper gives some arguments as to why these systems are recommended, how they will be supported and shows how they may be used together harmoniously using SGML.

Some additional benefits of SGML, such as the creation of a preprint database and the electronic submission of physics articles to publishers are also discussed.

This document supercedes any previous versions that have appeared, although the differences with the document presented at the CERN MEDDLE meeting on 7 February 1989 are minimal.

---

1) CERN, Geneva, Switzerland
a) Technical student from Leicester Polytechnic, Leicester, U.K.
b) Now at: Hogere Technische School Enschede, Enschede, The Netherlands
c) Now at: Technische Hogeschool Delft, Delft, The Netherlands
d) On leave from: Institute of High Energy Physics, Serphukov, USSR

## Acknowledgements

The ideas developed in this paper are the result of many discussions about SGML with a large number of people both inside and outside CERN. We gratefully acknowledge the help and work of A. Berglund (ISO), M. Bryan (Yard Software Systems) and M. Cowlishaw (IBM), as well as all our colleagues who are helping us to try to get SGML off the ground at CERN.

This document contains the names of many vendors and products. We have tried to give references which are as complete as possible, including dates and version numbers.

Wherever the name of such a trademark appears, we ask the vendor to accept this paragraph as an acknowledgement.

# 1. Introduction

The current text processing situation at CERN is very confusing. The user community that needs word and text processing tools consists of administrative and managerial staff in the 13 divisions (about 400 users) and a large part of the permanent and visiting scientific staff (about 3000 users).

The divisional secretariats use a large set of incompatible word processing systems. Among others, products from Norsk Data, Wang, AES, Philips, Olivetti and Nixdorf are installed. On the other hand, the scientific user community mainly uses batch like text formatting systems such as Waterloo Script and $T_EX$ on the IBM mainframe and the DEC VMS cluster systems. At the same time, PC based word processors like MS Word, WordPerfect, Macwrite and Fullwrite are gaining in popularity at an almost uncontrollable rate. Desktop publishing systems, of which Pagemaker and Interleaf are the most important examples, are also becoming more and more accessible.

It is the mandate of the MIS unit to recommend an integrated text processing environment [1] by **reducing** the number of incompatible systems now in use at CERN, by **buying** commercially available systems wherever possible rather than developing these ourselves and by **supporting** existing official (ISO, ANSI) and industry (de facto) standards which will guarantee compatibility between the different platforms and with future systems [2].

## *1.1 Supported platforms and reduction of systems*

As described in [2], rather than recommend a preferred operating system or environment for text processing, we assume that there will be MacOS, PC DOS and OS/2, VM/CMS, DEC VMS and Unix systems at CERN, and that we must integrate the text systems on these platforms. Out of the various solutions proposed, divisions or clusters of users may select that particular solution which is most appropriate to their environment.

**Any text system running on a different operating system will not be supported by MIS. A phase-out plan for obsolete systems [1] will be provided this summer (target date: August 1989; manpower required: 3 person months).**

## *1.2 Use of industry standards to achieve integration*

Since there is no system currently available running on the platforms mentioned above which satisfies all of the requirements listed in [2] (a condensed version of these has been placed in Appendix A), we need an interchange format to convert from one system to another. The two ISO standards which were designed with the aim of allowing interchangeability of text at the source level between different text formatting systems are the SGML standard [3] and the ODA/ODIF standard [4]. The SGML standard seems to be the more suitable of these two for CERN (see Appendix B for an introduction to SGML and Appendix C for a brief comparison between ODA/ODIF and SGML).

Although the portability of text is not always required, there are many other reasons why SGML could be needed. A checklist of questions to answer before rejecting SGML is given in Chapter 2. The use of SGML adds value to documents by enabling not only their printing, but also for example their addition to databases and the electronic submission of articles to publishers.

More specifically, the reasons why SGML is needed in the physics world, where (LA)$T_EX$ [5] seems to be a *de facto* standard, are given in Chapter 3. It should be emphasized however, that we are not trying to replace $T_EX$ by SGML. $T_EX$ will remain available independently.

A very popular *de facto* standard for printers is the PostScript [6] page description language. As many text formatters are capable of producing PostScript output, and more and more printers are capable of interpreting this language, the question arises if one should not use PostScript as the interchange language. Again, the reply is that SGML permits many other applications of documents. The details of this question are addressed in Chapter 4.

The proposal for the introduction of a set of text processing facilities that are integrated by the use of SGML is given in Chapter 6.

**SGML should be used at CERN as the standard for interchanging text from one system to another. Any text system that is not capable of interfacing to SGML will not be supported by MIS.**

## 2. When should one use SGML?

Although in principle all text may be processed by an SGML system, in practice the machinery of SGML may be too heavy for many documents. If the answer to all of the following questions is **no**, use of the basic facilities offered by a word processing system such as MS WORD (on a PC or Macintosh) or of an electronic typewriter is recommended rather than SGML.

1. Does the document need to be portable (e.g. from Word to $T_EX$)?
2. Will the document be added to a database?
3. Will the document be kept after it has been read?
4. Will the document be updated later?
5. Does the document have a complicated structure?
6. Is the document longer than a few pages?
7. Does the document have to follow a CERN corporate style?
8. Is the user already acquainted with SGML or any existing SGML tool?

If the answer to any of these questions is **yes**, use of a system that is capable of producing files that conform to SGML should be given serious consideration. SGML allows us to do more with text than simply send it to the printer.

## 3. Why do we need SGML if one already has (LA)$T_EX$?

As many high energy physics institutes have (LA)$T_EX$ (which is public domain software and may be obtained free of charge), and $T_EX$ is portable one may argue that (LA)$T_EX$ should be used as a standard text formatter.

We stress that $T_EX$ will be supported at CERN (see Chapter 5) and that macros will be provided for those divisions or user groups who have specific style requirements. This chapter tries to make the case for the use of SGML in addition to $T_EX$.

### *3.1 SGML based input systems vs. $T_EX$ based input systems*

SGML is a standard which concerns the whole process of document production, not only the formatting part. Its structure allows for applications of text which would be beyond what may be done with a formatter on its own, such as the creation of an input system.

Although initially at CERN on VM and VMS the supported user input system will be direct coding in SGML, it is clear that the ultimate goal is an input system where the user is not aware of using either SGML or $T_EX$.

WYSIWYG SGML editors are becoming available; a WYSIWYG $T_EX$ system may be an impossibility, although one could create one for a specific set of (LA)$T_EX$ macros. It is unlikely however, that $T_EX$ input systems will become available for every set of (LA)$T_EX$ macros.

Of all the generic markup systems, SGML is probably the least user unfriendly to code in. This argument should not be the main one, as normally the input system should add the tags to the document without the user being aware of this, but it does imply that the user support will be easier on those systems which do not have SGML input systems.

### *3.2 Electronic submission of physics articles to publishers*

SGML is a standard vehicle to transport documents from one platform to another. Apart from interchange inside CERN however, an important benefit for our user community would be the electronic submission of physics articles to publishers.

Some journals are now indeed accepting input of articles in machine readable form (the American Physical Society's "Physical Review", Steven Wolfram's "Complex Systems", Wiley's "Electronic Publishing, Origin, Dissemination and Design" are some examples). These journals accept articles coded in various different (LA)$T_EX$ macro packages; a popular macro package is PHYZZX [7], which is used at SLAC for producing preprints. Resubmitting an article to a different journal could imply recoding in the macro package specified by that journal.

In trying to comply with (ever changing) layout specifications for conference papers, preprints and articles, our user community wastes an enormous amount of time and energy. Needless to say that this situation also puts a heavy load on people supporting the text processing systems in CERN. A standard (LA)$T_EX$ macro package should be adopted, or all publishers should agree to use SGML (and themselves translate into their own macro package).

It seems that SGML is now slowly being adopted by the publishing world [8], and indeed a pilot project between CERN, Elsevier Science Publishers (who probably account for more than

50% of CERNs' scientific publications [9]) and the European Physical Society is underway. This will permit the physics author to submit a file in SGML form to the publisher rather than in camera ready form.

As part of this pilot project, Elsevier are also designing a (LA)$T_EX$ macro package for a standard physics article. These macros will closely follow the structure of CERNs' proposed standard physics article (see section 3.3 and Appendix D on page 25), and they will be available freely in the scientific world. Use of these standard macros by authors who prefer input in $T_EX$ rather than direct coding in SGML will permit easy translation of these documents into SGML.

Although these attempts at standardisation on a given macro package are very important, it should be noted that $T_EX$ was never intended to be an interchange standard. Characters that are important for $T_EX$, such as '{', '}', '¬' and so on are notoriously troublesome when a file containing them is transferred across heterogeneous computer networks because their place in the various ASCII and EBCDIC code pages is not defined unambiguously. Secondly, one should note that certain macro packages are very deeply intrenched and it is doubtful whether users that are perfectly happy with say, PHYZZX, can ever be persuaded to use anything else. Both these arguments point towards the use of SGML as the interchange standard.

However, be it via $T_EX$ or SGML, the benefits to the physics community of a shorter publication process resulting from mutual agreements between publishers and authors are evident.

### *3.3 Registration of the CERN SGML tagset with ISO*

SGML is an ISO standard (the second most ever sold one) whereas $T_EX$ is not. CERN intends to register its SGML tagset in the way recommended by ISO and make its applications (including the $T_EX$ macros mentioned above) available free to any HEP laboratory. If necessary, agreements will be made with manufacturers so that this may be done.

A CERN standard document set (see Appendix D on page 25) is being established with the CERN user community. The document set contains a single, common CERN preprint structure (currently EP and TH divisions support at least 8 different types), which will be consistent with the database applications foreseen by Elsevier Science Publishers as well as the Scientific Information Service group at CERN. Divisional style differences are completely compatible with SGML and will be supported by the provision of macros. The benefits of such a simplification to the organisation should be clear.

The advantage of registering this standard tagset is that publishers will know what HEP documents look like and will be able to accept scientific articles marked up in the standard HEP way.

### *3.4 Database applications (a preprint database)*

SGML was specifically designed to create document database applications. With (LA)$T_EX$ it is not clear how this should be done in a general way. This is an important consideration for documents with a long lifetime, as one cannot always predict their future use. All data for the current CERN preprint database is entered by hand. The use of SGML would make the creation of tools to add preprints and other documents into this database much easier.

### *3.5 Portability of SGML documents vs $T_EX$ documents*

As remarked above, $T_EX$ was not written with the intention that it would be an interchange format that would survive a filetransfer across various heterogeneous networks. Another problem is caused by the fact that (LA)$T_EX$ files often contain information about the fonts which are needed at printing time. Although DVI files are independent of the output device, they are created using width tables of fonts which are presumed to be present when the DVI file is processed for printing. Everyone has experienced the ■'s that occur when the fonts are not present at the receiving end.

SGML makes a document completely independent of the output system, including the printer.

### *3.6 Creation and maintenance of macros*

The use of an SGML input system implies that the underlying macros (they could be (LA)$T_EX$) may be written in a very simple way, as most of the checking will take place when the document is processed by the parser. This will lower the necessary support for the macros and require less time to write them.

### *3.7 SGML and the future office system*

SGML will be chosen by the future office system [2] at CERN, as it seems the only way to harmonise the many existing text processing systems currently in use at CERN.

**Use of a standard (HEP) set of $T_EX$ macros (permitting easy conversion into SGML) or direct coding in SGML (with translation into $T_EX$) will combine the benefits of the popularity of $T_EX$ with the standardisation offered by SGML.**

## 4. Why do we need SGML if one already has PostScript?

PostScript is a popular page description language which is being adopted by many manufacturers of laser printers. There are also a number of text processing systems which directly produce output in this language. It is a programming language intended to describe documents in their final form.

If only transportability of final form documents to a remote laserprinter is required, the use of PostScript will be adequate (provided there are no problems with the translation of special characters across heterogeneous networks). However, if one wants to do more with a document (edit, add the file to a database), a system such as SGML is necessary. One should remember that the use of SGML makes a document completely independent of any text formatter or output system. There are still many printers that do not accept PostScript.

**We recognise the importance of PostScript and will support it wherever possible. This implies that PostScript interpreters will be provided such as the PostScript interpreter for the IBM 3812 printers [10] and DVI to PostScript translators for $T_EX$.**

## 5. Operational model of text processing systems at CERN

In theory, the use of SGML should guarantee not only portability but also ensure a consistent user interface across different platforms (see the subsequent chapters). It seems therefore reasonable to try to install SGML based input systems everywhere. However in practice, there are very few of such input systems on the market at a reasonable price, particularly on (IBM compatible) PCs and mainframes. Awaiting more SGML products we propose initially to only use SGML as a conversion standard between the text/word processing systems[1] described in the following sections. It should be clear that the emphasis is at all times on the purchasing of systems, but that some local developments may be necessary to allow the integration.

A very important local development is the SGML application interface [11]. This application interface is necessary to permit the translation of SGML tags into a formatting language or another application (in a machine independent way). A local one had to be developed since not all the existing SGML application languages give us sufficient access to the structure of the document. Furthermore, on certain platforms there are no application languages available or they are too expensive; those that do exist are all different and very machine dependent.

The second stage will commence when more SGML input systems are available. We will then be able to provide WYSIWYG SGML systems, database applications and hypertext systems (see [12]). Gradually all the requirements of text processing systems mentioned in Appendix A will be met.

### *5.1 Word processing on a PC running DOS or OS/2*

MS Word [13] has been proposed by MIS as the interim recommendation [14] for a PC based word processing package. For the creation of most office documents (letters, memoranda, minutes and agendas) the use of MS Word will be adequate. However, sometimes documents will have to be exchanged with other systems, and sometimes documents coming from other systems will have to be modified using MS Word. The proposal is to achieve this via conversion to and from SGML.

Conversion of MS Word PC documents to SGML (see [15]) may be achieved via a set of stylesheets that are structured similarly to SGML layouts (or DTDs). For ease of use, a set of macros has been written that automatically attach these stylesheets, and prompts the user through the structure of a document. The addition of SGML tags is done by a simple conversion program. Another advantage of using stylesheets is that all documents produced with them will have a similar style. Any document made without these tools will not be guaranteed to be portable.

A distribution diskette containing these macros, stylesheets plus conversion program with the accompanying documentation [16] is available from the authors. Note that the stylesheets and conversion program are written specifically for the currently existing SGML implementation on VM. For the moment neither mathematical formulae nor tables are supported.

MS Word has facilities to integrate graphics in PostScript form, but none for generating mathematical formulae. At present, there is no good PC based mathematical formulae editor although it is our intention make a recommendation **(target date: December 1989; manpower required: 1 person month).**

---

[1] Note that these systems do not (yet) satisfy all requirements listed in Appendix A on page 18.

The translation of SGML into MS Word will for the moment have to be done on the Macintosh, where a mapping of the standard CERN document set onto RTF [17] will be provided ( **target date: June 1989; manpower required: 2 person months)**. Use will be made of the SGML application interface mentioned above.

Users with an immediate need for high quality mathematics text should use PC $T_EX$ [18]. A facility will be provided to translate an SGML document into PC $T_EX$ or to submit a batch job to VM to format the SGML document there. A distribution diskette containing the recommended $T_EX$ macros for the CERN standard document set will be prepared **(target date: September 1989; manpower required: 1 person month)**.

**We propose to remove the interim status of the recommendation of MS Word as the standard PC word processing package. Support to PC Word will be given through the stylesheets, macros and templates provided by MIS.**

For a pictorial representation of text processing on PC's, see Figure 1.

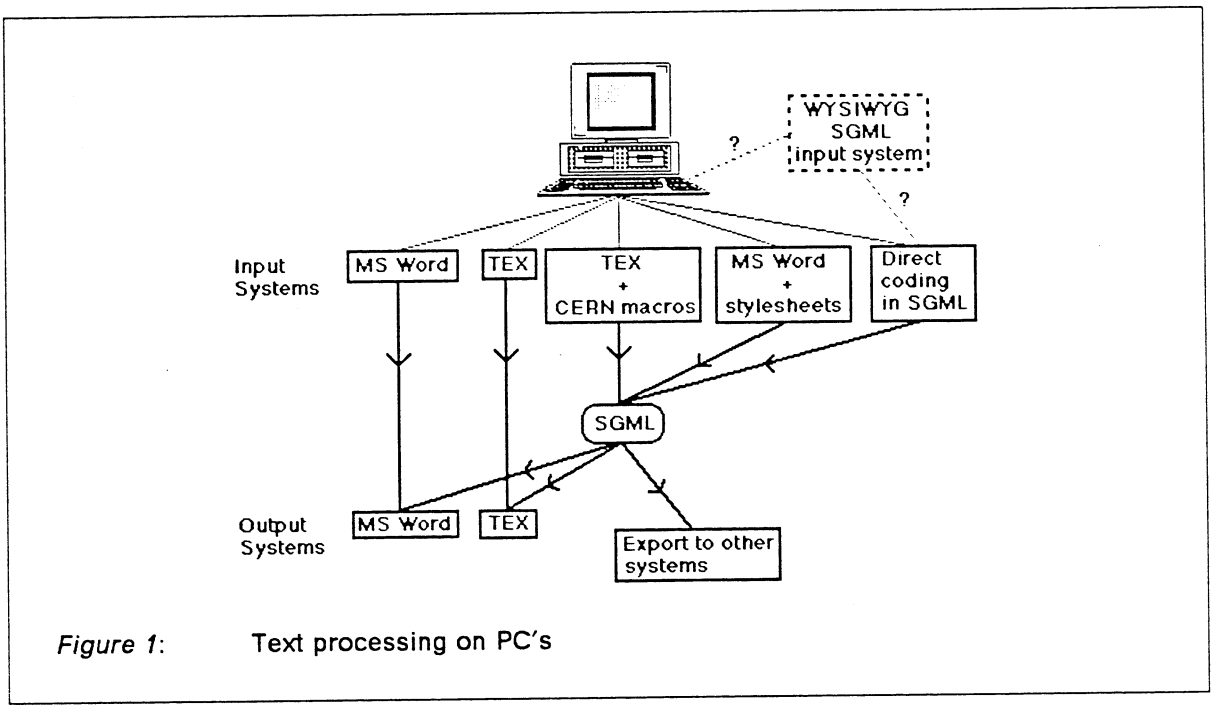


*Figure 1*: Text processing on PC's

## *5.2 Word processing on a Macintosh*

MS Word [19] has been proposed by MIS as the interim recommendation [14] for a Macintosh word processor. As in the PC case, for most purposes MS Word by itself will suffice. If needed, the conversion of MS Word documents to SGML may be achieved by providing a set of macros and stylesheets. Unfortunately version 4.0 of MS Word for the Macintosh does not have a macro facility that is powerful enough to achieve a similar input system such as the one provided for PCs.

However, stylesheets for the standard CERN document set will be provided **(target date: June 1989; manpower required: 1 person week)**, and the conversion to and from SGML will be achieved via RTF **(target date: July 1989; manpower required: 1 person month)**.

The SoftQuad Author/Editor [20] system will be recommended as SGML input system. Support for tables is much better in the next version, which apparently will be in $\beta$ test in September 1989. A diskette will be made available by the authors containing compiled DTDs of the standard CERN document set for use with Author/Editor **(target date: August 1989; manpower required: 1 person month)**.

The **Mathtype** [21] system may be used as in interactive mathematical equation editor with MS Word; it will also be integrated with Author/Editor as well as produce output in an SGML compatible format **(target date for version 2.0: June 1989; manpower required: none)**.

It will be possible to translate an SGML file (that comes out of Author/Editor for example) into TEX [22] **(target date: September 1989, manpower required: 1 person month)**, MS Word, or to submit a batch job to VM to format the SGML file there.

**We recommend MS Word as the standard word processor for Macintosh. Support will be through stylesheets. As SGML input system we will support and recommend the SoftQuad Author/Editor system.**

For a pictorial representation of text processing on Macintoshes, see Figure 2.

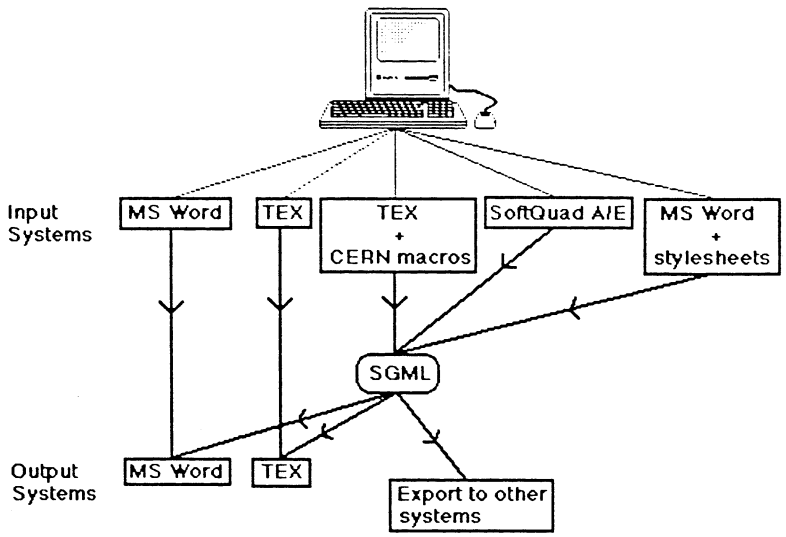


*Figure 2:* Text processing on Macintoshes

## *5.3 Text processing on VMS*

The text processing facilities on VMS will be based on TEX. There will be a conscious attempt to ensure compatibility between all TEX systems that are offered on other platforms, particularly the service offered on VM.

An SGML system based on TEX will be made available soon **(target date: June 1989; manpower required 1 person month)**. This system will use the Hahn-Meitner SGML parser [23] and will be be 95% compatible with the current SGML system on the IBM. This obviates the need for

sending remote batch jobs to the IBM for processing SGML files that were created on the VAX (the current resource consuming procedure).

Work on a standard CERN (maybe HEP) TEX macro package is in progress. This macro package will be freely available from us **(target date: September 1989; manpower required: 1 person month)**. The macros from Elsevier Science Publishers for standard physics articles will obviously have a great impact in our community. However, the most commonly used macro packages such as PHYZZX, LaTeX, the American Physical Society macros, the Institute of Physics Publishing macros etc will also be made available on a "use at your own risk" basis.

A full SGML system, based on a parser (probably [24]), the above mentioned macro package(s) and the SGML application interface will be installed by end of the summer **(target date: September 1989; manpower required: 2 person months)**. A TEX to SGML conversion program will have to be written **(target date: October 1989; manpower required: 1 person month)**.

In addition, support (installation and/or purchase if necessary) will be offered for the postscript printing tools PSDVI and PSPRINT, which will also permit TEX output to be printed on local postscript printers. Finally, a selection and possibly installation of previewers is foreseen **(target date: December 1989; manpower required: 3 person months)**.

**Support for TEX on VMS will be given through the standard CERN TEX macros and SGML.**

For a pictorial representation of text processing on VMS, see Figure 3.

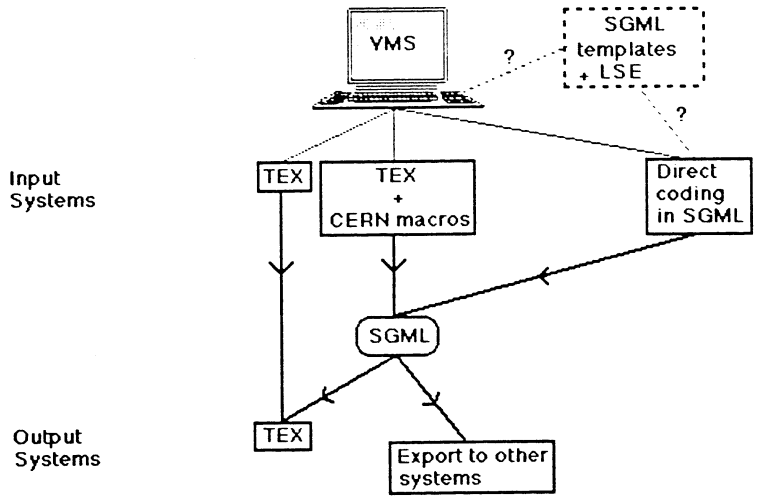


*Figure 3*: Text processing on VMS

## *5.4 Text processing on VM/CMS*

The existing SGML system based on Waterloo Script [25] will be replaced by the IBM SGML to DCF translator program [26] together with IBM DCF (Document Composition Facility) [27]. A production service based on DCF is planned **(target date: September 1989; manpower required: 6 person months)**. The work to be done includes installing the system, writing of macros, providing command files etc. The existing SGML implementation will be frozen. Waterloo Script will be kept for an undefined period to allow old documents to be printed.

IBM has firmly committed itself to SGML and has several other products for use with DCF, known as the -master series. These will also be ordered and be made part of the production service **(target date: September 1989; manpower required: part of the 6 months above)**.

The PostScript to "IBM printer datastream" format convertor [10] has been ordered. This will allow printing of PostScript documents on the many 3812 laserprinters on site at CERN **(target date: August 1989; manpower required: 1 person month)**.

A system to aid with the production of SGML documents based on the context sensitive LPEX editor [28] will be provided by us **(target date: pending; manpower required: 1 person month)**.

$T_EX$ will be available on the same basis as it is for VMS.

Note that users that do not require SGML may directly use DCF or $T_EX$ in the way that they are used to. If they require consistency with SGML, they may use the macros mentioned above.

**Support for all text formatters on the IBM mainframe system will be through SGML.**

For a pictorial representation of text processing on VM/CMS, see Figure 4.

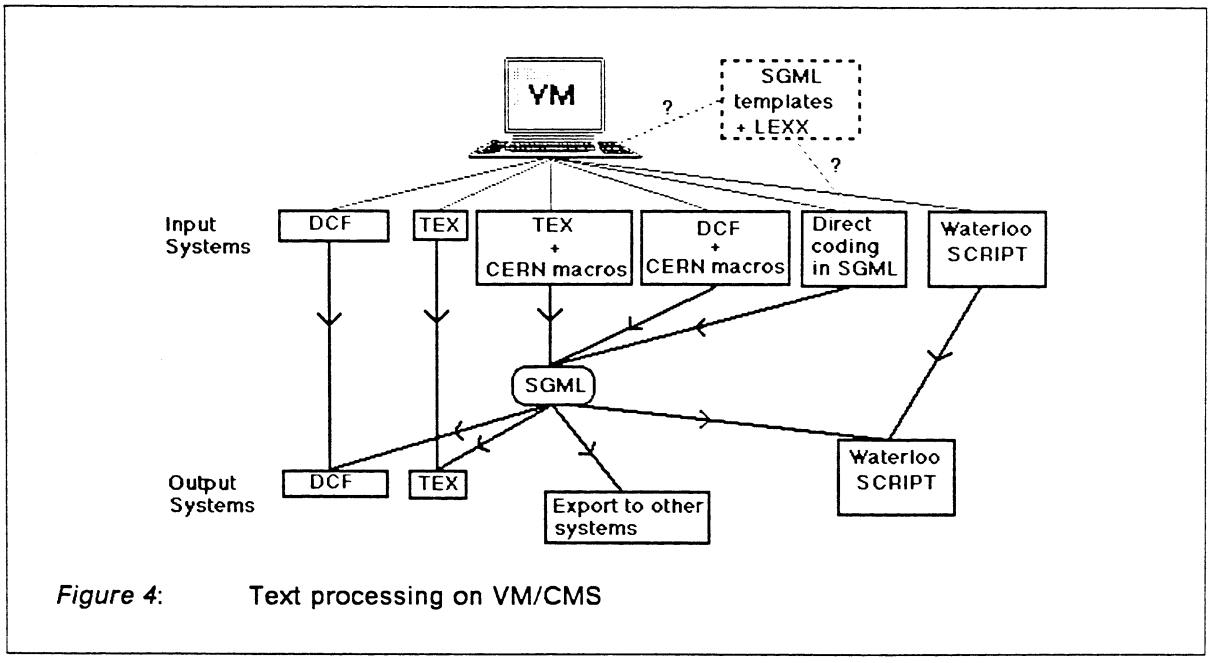


*Figure 4*: Text processing on VM/CMS

## 5.5 *Desk top publishing on Unix (Apollo/SUN) and VMS (VAXstation) Workstations*

Although this user community is not very large at this moment in time (200-300 Apollo's and Vaxstations), the total number will potentially grow to around 1200 in the early 1990's (see the Final Report of the Working Group for Experiments [29]). It is therefore important to anticipate a good system at an early stage in the evolution of this group of users.

For users wishing a batch product, the TEX product from Arbortext (plus CERN macros if compatibility with SGML is required) and previewer will probably suffice.

For users wishing a high quality WYSIWYG text formatting system, we recommend the use of Interleaf TPS [30]. Interleaf is firmly committed to SGML. Conversion of Interleaf TPS documents to SGML will be possible from release 4.0 onwards, and full support of SGML will be given from release 5.0 onwards. MIS will give indications on how the migration to and from SGML may be done. A PC version of Interleaf exists (TPS version 3.0) and will be evaluated. A Macintosh version also exists (TPS version 3.5), but requires a large amount of RAM (8 Megabytes) and an 80 Megabyte hard disk.

**Support will be given to TEX initially and later to Interleaf as CERNs' recommended desk top publishing package for high-end workstations running UNIX (Apollo/Sun) and VMS (Vaxstation). In view of the large number of potential users a site wide license should be negotiated with Interleaf.**

For a pictorial representation of text processing on workstations, see Figure 5.

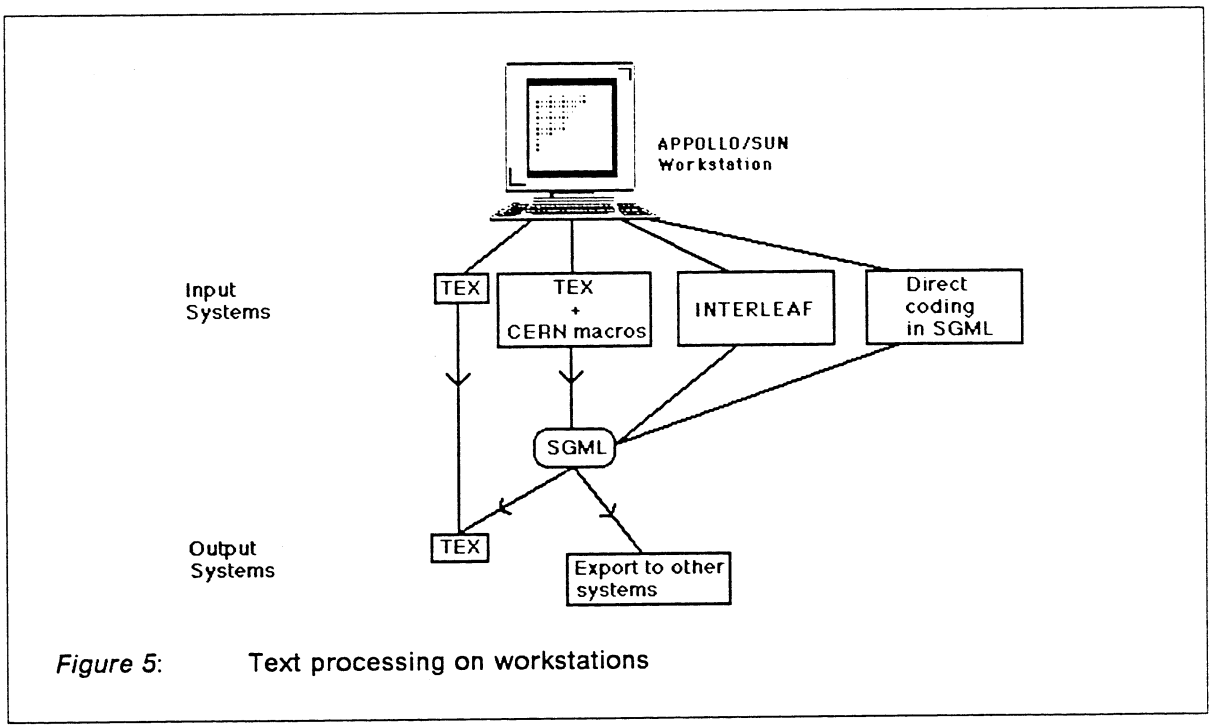


*Figure 5*: Text processing on workstations

### *5.6 Mainframe text processing using a PC as a terminal*

To be able to use the PC for interactive text processing on either the IBM or DEC systems, the TCP/IP terminal emulator must contain emulation for IBM 3179 graphics terminals or Tektronics 4014 terminals. This is because the IBM previewer (Browsemaster for DCF [31]) expects a GDDM datastream and the TEX previewer on the VAX expects a Tektronics 4014 datastream. Unfortunately, so far, our TCP/IP PC emulator [32] does not support graphics terminals, and consequently no previewing will be possible.

One alternative would be to transfer a DVI file (or its equivalent) to the PC and use a local previewer. In any case a dumb terminal datastream will have to be provided for both TEX and DCF (**target date: September 1989, required manpower: 1 person month**).

### *5.7 Mainframe text processing using a Macintosh as a terminal*

The same remarks as given for above for PCs apply, but since the Macintosh has a much better graphical interface, a good 3179G emulator is available [33]. The Macintosh may therefore be used to preview formatted output from the IBM.

### *5.8 Future systems*

Choice of the SGML standard implies that in principle any future SGML based input system should be compatible with the systems proposed above, and such a future system should be easy to install and support by MIS. For example, the Interleaf products may one day be run on all platforms in an SGML compatible way.

## 6. Proposal for SGML installation at CERN

The operational model for text processing at CERN described in Chapter 5 will be the first step towards a full SGML installation. The advantages of everyone being able to use the text processing facilities on their favourite system and at the same time being able to exchange the data with people on other systems are obvious. Although this plan tries to minimise the amount of work to be done at CERN, we must, before using any part of the system individually, do the following:

1. Identify a standard CERN document set with their elements.

2. Define their structure in terms of the SGML language by designing the DTDs (Document Type Definitions, see Appendix B).

3. Write macros/stylesheets for the standard CERN document set corresponding to a standard CERN default style.

4. Write WORD to SGML convertors.

The following points are part of the second stage of the SGML installation:

5. Install WYSIWYG SGML editors.

6. Create database applications.

7. Create hypertext systems.

As we try to rely on bought rather than homegrown software, the final result should be easier to maintain than at least the present text processing system on VM, which relies heavily on local modifications to Waterloo Script.

## *6.1 A standard CERN document type set*

Together with the Composition and Printing Group, the Editorial Group, the Scientific Information Service and the divisional secretariats, we propose that the following set of documents should be included in a standard document set for SGML:

1. Letters.
2. Memoranda, Technical Notes etc.
3. Minutes/Agendas.
4. Preprints (LEP, TH, EP, ??), and Conference Papers. These are documents which normally would be distributed outside CERN.
5. Reports (Divisional). These are documents which normally are for distribution inside CERN.
6. Books, Guides, Writeups, Manuals.
7. Transparencies.
8. Yellow Reports.
9. The Weekly Bulletin.
10. Newsletters.
11. Seminar announcements.
12. A FAX cover page to be used for all document types.

Approval for this CERN standard document type set and its elements (which are described in appendix D on page 25) is now being sought with the user community and secretarial staff.

Please note that it is not our intention to create document styles that will be fixed forever. Types may be added and modified as required. The documen types should be flexible and not stifle the users personal creativity. A set of visually attractive presentation styles could be created with the help of a graphic artist, if the organisation considers this aspect important.

## 6.2 Structure of the DTDs

The DTDs will be designed in a very flexible way, permitting the use of SGML as a word processor. The aim is that eventually the user will have full control over the page. An attempt will be made to conform as much the SGML standard as possible, whilst also remaining backwards compatible with the currently existing SGML "notation" (which does not conform fully to SGML) on CERNVM.

To be more specific, the following two features (which will meet the most commonly encountered criticism of SGML) will be added to our DTDs during the first stage:

1. A **no-layout** DTD. This is a layout without any specific title page, where all tags normally occurring in the main body of the document may be used.

2. A limited set of **specific** processing instructions will be supported for creating new pages, new lines and empty lines. SGML foresees a tag for this purpose; our import and export tools will translate the supported set from one text formatting language into another. See [12] for more details.

A second stage, where many possibilities of the SGML standard will be exploited, will be proposed at a later date. For a full technical description, see [12].

## 6.3 The DTDs

The DTDs corresponding to the standard CERN document set may be found in [12].

## 6.4 Macros/stylesheets

- Stylesheets for each document in the standard CERN document set are available for MS Word on the PC [16].

- $TEX$ macros are being written for each document in the standard CERN document set. This will permit the translation from SGML into $TEX$, and their use will facilitate the translation from $TEX$ into SGML.

- DCF macros will have to be written for each document in the standard CERN document set.

- An SGML input system using LPEX is being developed, initially for use with VM. It may later be ported to the PC.

## 6.5 Database applications

We plan to build an SGML-SQL (Oracle) interface described elsewhere [41]. This interface could be used for creating a preprint database, applications such as the CERN program library manual and CERNDOC, CERN's document filing and retrieval system. Again, the SGML application interface [11] will be invaluable.

## 7. Conclusions

We have given a proposal for an integrated text-processing environment on CERNs' most important computing platforms, i.e. VM/CMS, VAX VMS, MS-DOS (and later OS/2), MacOS, and Unix.

By using SGML as an underlying format for text interchange, many additional benifits will be obtained. For example, by agreeing with CERNs' principal publishers on a common scientific article format, physicists will be able to electronically submit their articles. This could substantially reduce the time needed for the publishing process. Another advantage of SGML is that it would permit us to easily construct a preprint database at CERN.

The proposal described in the above, although it may appear ambitious, is based entirely on purchased (or freely obtainable) products. We believe it is the minimum configuration for CERNs' needs.

The approximately 21 person months of manpower required should be available during the first half of 1989. The second stage of the SGML implementation may then commence.

# Appendix A

## Requirements for text processing systems

This appendix contains the text processing systems requirements as defined in [2]. It is very important however to remember that text processing should be well integrated with the other office systems components.

1. The text processing system should be able to produce high quality output with the following properties:

   a. Accented characters as required by the member state languages (including Scandinavian) at CERN should be available.

   b. All characters should be available in at least 4 pointsizes (8, 10, 12 and 18) and 4 presentations (roman, bold, italic and bold italic text)[2]. The symbols listed above should be available.

   c. Mathematical and scientific symbols, superscripts and subscripts and greek letters should be available.

   d. Proportionally and mono spaced text.

   e. It should be possible to position footnotes at the bottom of the page where they are required.

   f. The system should be able to generate indices, tables of contents and more generally be capable of referring to events that occur during the execution of the text processor (allowing the creation of pointers to page numbers and so on).

   g. The system should support multiple column output.

   h. The system should be able to generate forms overlays for all the standard CERN stationery.

   i. It should be possible to request tables to be rotated 90° to permit inclusion of longer tables in documents.

   j. The system should be capable of producing A3 (in addition to A4) output.

   k. The system should be able to produce output in the PostScript page description language.

   l. The system should provide a capability for processing mathematical formulae, with the following properties:

---

[2] This is a constraint on the ouput device, which must have these fonts available. Note however, that the widths must be available to the text formatter (to enable formatting) and that different fonts should be available for all sizes and presentations (scaling characters up is not acceptable).

i. It should be possible to print formulae everywhere in the document, including in titles, chapter headings, tables and figures.

ii. It should be capable of producing complicated mathematical formulae such as shown below (see formula (1)). Note that the formula is numbered, and that the size of the brackets is adjusted automatically.

iii. The mathematical formula processor should have an built-in knowledge of mathematical typesetting rules, i.e. automatically change the font for limits of integrals, print the integrand in italic characters and so on

iv. The system should be capable of producing large brackets for printing matrices. It should not be necessary to scale the brackets by hand when the matrix is modified later.

v. For (multiple or nested) fractions it should be capable of positioning numerator and denominator correctly.

vi. It should be simple to number formulae and to refer to them elsewhere in the document.

vii. An interactive formula entry system should be available to guide the user in entering limits of integrals, numerators denominators and so on.

$$\Pi^{G^2}_{\mu\nu}(q,m_1,m_2)=\left[\frac{i}{(2\pi)^4}\int d^4p\ \mathrm{Tr}(\gamma_\mu S_G(p,m_1)\gamma_\nu S_G(p-q,m_2))\right]_{G^2}$$

$$=\frac{1}{24}<\frac{\alpha_s}{\pi}G^2>\left\{\int dx\, x(1-x)\left[(\frac{3}{C}+\frac{x(1-x)}{C^2}q^2)g_{\mu\nu}+\frac{2x(1-x)}{C^2}q_\mu q_\nu\right]+\right.$$

$$x^2\left[\left(-\frac{3m_1m_2}{C^2}+\frac{2m_1^3m_2}{C^3}+\frac{m_1^2}{C^2}x\left(1+\frac{2}{C}x(1-x)q^2\right)\right)g_{\mu\nu}-\right. \qquad (1)$$

$$\left.\frac{4m_1^2}{C^3}x^2(1-x)q_\mu q_\nu\right]+(1-x)^2\left[\left(-\frac{3m_1m_2}{C^2}+\frac{2m_1m_2^3}{C^3}+\right.\right.$$

$$\left.\left.\left.\frac{m_2^2}{C^2}(1-x)\left(1+\frac{2}{C}x(1-x)q^2\right)\right)g_{\mu\nu}-\frac{4m_2^2}{C^3}x(1-x)^2q_\mu q_\nu\right]\right\}$$

2. On the input side, the following requirements may be formulated:

a. The system should provide good tools for easy production of tables.

b. The system should be well integrated with address lists and other database facilities.

c. Inclusion of graphics at any desired place in the text. It should be possible to place text next to the graphics. The graphics may come from GKS, CGM, PostScript or be output from any of the programs of the *future office system*.

d. The system should have a spelling checker with English, American and French dictionaries.

e. Inclusion of images in the text as required. The same requirements hold as for graphics. Here we are thinking of scanned images (i.e. line drawings and half-tones) in particular.

f. Interfaces with computerised translation facilities should not be forgotten, but are probably not mandatory immediately.

g. Voice annotation would be interesting but probably not mandatory. It should be remembered that this could have consequences on network traffic (the equivalent of an A4 page of text, i.e. 4K bytes as ASCII, is 400K bytes as an uncompressed bitmap, but 1.2M bytes as digitised voice).

3. A text processing system will be very closely linked to the editor on the particular system used for entering the text. In this area we place the following requirements:

   a. The system should have a WYSIWYG facility for text (including accented characters), graphics, mathematics and tables. This probably poses additional requirements on the hardware used, as viewing graphics and mathematics require a bit-map screen plus a reasonable amount of local processing power to unscramble (possibly compressed) bit-maps. It should be realised however, that true WYSIWYG will be very hard to achieve, and is probably unnecessary in the case of a good formatter.

   b. PostScript editors and/or screens are desirable.

   c. If a decompressor is present, this should be able to handle the CCITT group IV compression standard.

4. To provide the needs for corporate publishing the following is required:

   a. Style sheets are required to permit the definition of CERN defaults.

   b. The system should be able to accept generic (perhaps as well as specific) input. This has the advantage that many boring tasks (index, table of contents) are performed automatically. Documents that are marked up in a generic form are easier to maintain, and no requirements of typographic skills are required of the user. Note that this requirement is not contradictory to the WYSIWYG requirement.

5. Full support of the SGML standard is required for the following reasons:

   a. To facilitate the exchange of documents between different systems.

   b. To produce documents that have a long lifetime, and that will probably be worked on by multiple authors.

   c. To produce documents that have a complicated structure.

   d. To produce corporate publishing functions.

6. To summarise, the system should provide at least everything we can get today on a Macintosh (or any existing system currently in use at CERN) and probably more.

# Appendix B

# What is SGML?

## *B.1 Elements, tags and attributes*

SGML [3] is the ISO standard for marking up text. The underlying assumption is that text consists of logical components which are called "elements". The standard specifies the manner in which the elements of a document should be indicated in the text. Marking up is done via so-called *tags*. For example, a paragraph element would be indicated by a paragraph *starttag* **<P>**, followed by the text of the paragraph itself, and concluded by a paragraph *endtag* **</P>**. "Attributes" may be given on a tag to change default values. For example, an identifier attribute **<P ID=XYZ>** may be added to the paragraph to permit a reference to it elsewhere in the document.

To ensure complete generality, the standard does not specify tagnames. In a sense this is unfortunate since it implies that a definition of the tags and the structure of the document is needed to guarantee portability; on the other hand it does allow the user complete freedom to decide on the types of documents SGML should be used for.[3]

## *B.2 The Document Type Definition (DTD)*

The key idea of SGML is to **separate** the contents and the structure of a document from the way it is processed, for example by a formatter. Achieving this makes the document independent of the underlying formatting system and improves its portability. The structure of a class of documents (e.g. reports) is defined by a "Document Type Definition". The DTD mainly contains the following information:

- The names and definitions of all tags that may occur in a document of that class.
- How often they may appear.
- The order in which they must appear.
- Whether the end-tag may be omitted.
- The contents of the element delimited by the tag, i.e. the names of the other elements that are allowed to appear inside it, down to the character data level.
- The attributes of the tag (c.f. style sheets for a word processor).
- The reference symbols that may be used inside the text.
- Any context sensitive properties of the text (e.g. the DTD could be set up in such a way that a blank line followed by a line which is indented 3 spaces is automatically assumed to be the start of a paragraph).

---

[3] Note that ISO 9070 - Registration Procedures for Public Text Owner Identifiers, would permit CERN to register its tagset with an independent registration authority, probably ANSI [?].

In a complete SGML application, any user is free to create any DTD that is required. To be portable however, the DTD should be present at the receiving end.

## *B.3 SGML input systems and other SGML applications*

The SGML standard defines rules for the formal construction of DTD's via a "meta markup language". Although the ordinary user normally does not need to know anything about a DTD, this approach permits text to be manipulated beyond its conventional purpose, the printed sheet of paper. SGML permits the construction of "hypertext" systems, but could also be used to create links between text formatters, "Electronic Document Interchange" [34] and databases applications. See [35] for a description of the operational model of the various SGML related standards.

Having the structure of a class of documents in a separate file permits the design of an **SGML input system**, or a structured document editor (see Figure 6 on page 23). The editor will read the DTD and know which tags are allowed at all points in the document, or recognise any structure already present in the document and add markup tags accordingly.

An example of a commercially available SGML input system is the SoftQuad Author/Editor product for Macintosh [20] (see section 5.2 on page 9). Several research projects concerning interactive structured document editors exist, amongst which the most promising, in our opinion, is the Quill editor [36]. Context sensitive editors such as LPEX (IBM) [28] and LSE (DEC) [37] may be adapted to become SGML input systems. One could also use a PC based word processor (e.g. MS Word, together with some well defined macros and style sheets) as input system and use the context sensitive tag adding properties of the SGML parser to add tags automatically if a document in SGML format is required later.

Note that the input system may be physically located on a different computer from the output system.

## *B.4 SGML parsers*

Typically, a DTD is given to a *parser* or *rules builder* to be *compiled*. This means that the DTD is checked for consistency and conformance to the SGML standard. The compiled DTD or the set of rules that correspond to the DTD are given to the input system and form a DTD specific parser. With some parsers, the DTD can be left in un-compiled form.

Once a document has been marked up conforming to the SGML standard, the DTD specific parser (if the DTD was compiled) will check that the document conforms to the structure defined by the DTD; if the DTD was not compiled, the parser will analyse the DTD (which must then be part of the document) and check whether the text conforms to the structure defined by the DTD. An example of a commercially available SGML parser recently purchased by CERN is the SOBEMAP parser for PC [38]. The Hahn-Meitner institute have given us their SGML parser (for VMS) which is being used by the Deutsche Forschungs Netz [23]. IBM has announced a parser for VM [26] parsers for UNIX exist [24] and will be purchased soon. Interleaf have announced that a future version of their desktop publishing package will include an SGML parser [39].

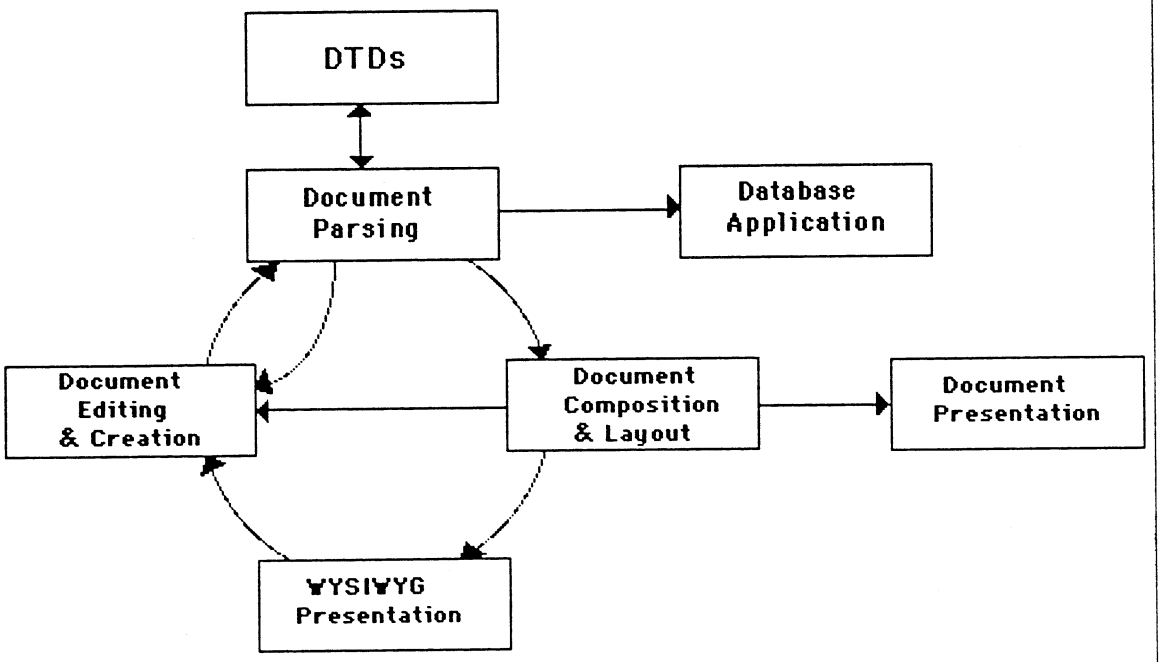


*Figure 6:* The parts of an SGML application

## *B.5 Output systems, text formatters*

A document which has been succesfully parsed may be sent out to an output (text processing or database) system. Please note that such documents are completely independent of any component in the SGML application (including the input system which could be MS Word, and the output system which may be TEX). Examples of text processing systems which may be used to process SGML documents are Waterloo SCRIPT (available at CERN on VM) or TEX (available at CERN on all platforms). Database applications for use with Oracle [40] are being planned [41]. Another application would be the formatting of the document for the screen to allow previewing and the creation of a WYSIWYG system.

## *B.6 Summary of a full SGML application*

To summarise, a full SGML application thus comprises:

1. A set of **DTD's**.
2. An **input** system.
3. A **parser**, or **rules-builder** to read (compile) DTDs' and to produce a DTD specific parser.
4. An **output** system:
   - A **text-formatter**.
   - A **database** application.

# Appendix C

## A comparison of SGML with ODA/ODIF

Another ISO standard for text interchange is ODA/ODIF [4]. A very brief comparison between these two standards can be summarised as follows:

1. ODA/ODIF was designed to communicate the author's intentions with respect to the presentation of the document, whereas SGML decouples the appearance of the document from its logical structure. Although Appendix E of Part 5 of this standard describes the *Office Document Language*, an SGML conforming **application** containing the rules for using the SGML Document Interchange Format [42] for ODA/ODL documents, one cannot use this as a means of converting ODA/ODIF documents into SGML. It is very difficult to glean information about the structure of a document from information about the way it is formatted.

2. It would therefore seem difficult to write application programs that can profit from a structured, well-defined document description which would be needed for the creation of database and hypertext systems. Indeed, whilst the SGML source form of a document is perfectly readable, the same cannot be said for its ODA/ODIF equivalent.

3. The scope of ODA/ODIF is office documents, whereas SGML more generally addresses any kind of structured text.

4. ODA cannot at present represent datatypes like tables and mathematical or chemical symbols.

5. The ODA/ODIF standard is very large and complicated, and the number of available conforming products is not very great. To create an ODA/ODIF document without a corresponding product is practically impossible, and in view of the large number of different systems in use in the physics author community ODA/ODIF does not seem a suitable choice.

# Appendix D

# Elements of the CERN standard document type set

This appendix contains the elements of a set of standard CERN document types. This list is not intended to be static; document types may be modified and others may be added to the list later.

## *D.1 Elements of letters*

For letters, we identify the following elements:

1. The official CERN letterhead. Other letterheads (provided they are for official use) will be supported through an attribute on the letter tag.
2. The name and address of the recipient of the letter.
   a. Family name(s)
   b. First name(s)
   c. Title-1
   d. Title-2
   e. Institution
   f. Address-1
   g. Address-2
   h. Prefix
   i. Town, Postal Code
   j. Country
3. The telephone number of the sender of the letter
4. The fax number of the sender of the letter
5. The electronic mail identifier of the person sending the letter
6. The reference number of the sender of the letter
7. The reference number of the recipient of the letter
8. The date
9. The attention name
10. The subject of the letter
11. The opening salutation
12. The basic document components
13. The closing salutation
14. The name and address of the sender of the letter
15. Additional distribution headings (PS, Enclosures, CC)
16. Distribution lists per heading

All the elements are sequential and should appear only once (except for the additional distribution elements at the end). They should only contain character data.

## *D.2 Elements of memoranda*

1. The official CERN memorandum heading.
2. The date.
3. A reference number.
4. The distribution headings (To, From, Copies etc.)
5. Distribution lists per heading. Very long lists should be displayed on a separate page.

6. The subject.
7. The basic document components.
8. The name of the author.

The elements before the basic memo components may appear in any order and may appear an arbitrary number of times. They should contain only character data.

## *D.3 Elements of minutes/agendas*

1. One or more title lines
2. Confidentiality of the meeting
3. A reference number
4. Name of the group holding the meeting/agenda. The original language should be an attribute
5. The date of meeting-1
   a. The place of meeting-1
   b. The subject of meeting-1
   c. The distribution headings (Present, Absent, To, Invited etc.)
      - Distribution lists per heading.
6. The date of meeting-2
   a. The place of meeting-2
   b. The subject of meeting-2
   c. The distribution headings (Present, Absent, To, Invited etc.)
      - Distribution lists per heading.
7. The basic document components.
8. The name of the secretary (trailer)
9. The meeting number

The elements before the basic minutes/agenda components may appear in any order and may appear an arbitrary number of times. They should contain only character data.

## *D.4 A standard document type*

This document type should be used for: conference papers, divisional reports, books, guides, writeups and manuals.

1. The Article. Attributes for indicating language, (experiment) code, security, status and version number. Graphics, mathematics and index entries may appear anywhere in the document.
   a. Publication information (e.g. Nucl. Phys. B166 (1980) 256)
   b. Copyright notice
   c. Keyword list
   d. Front material
      i. The title page
         1) One or more title lines. The possibility to indicate a short title.
         2) The date. Possibility to indicate 'received', 'revised' or 'published'.
         3) One or more reference numbers
         4) A general note. Also for updates on earlier versions of the same paper.
         5) Conference information.
            - Conference code.
            - Conference title.
            - Conference date.

- Conference place.
- Conference prefix: "Presented at:"

6) Journal information
   - Title of the journal.
   - Publication prefix: "Submitted to:"
7) Book/proceedings information.
   - Title of the book/proceedings.
   - The name of the editor of the book/proceedings.
   - The name of the publisher.
   - The place of publication.
   - The date of the publication.
8) A list of authors
   a) A name per author (one mandatory)
      i) Initials per author
      ii) Surname per author
   b) A note per author ("Now at:...")
   c) A line per author
9) A list of addresses
   a) Name of the department
   b) Name of the institute (one mandatory)
   c) Institute qualifier
   d) City
   e) State/County
   f) Country
   g) Postalcode
10) Collaboration name

ii. The abstract (may contain basic document components)
iii. The preface (may contain basic document components)
iv. Acknowledgments (may contain basic document components)
v. The table of contents (not keyed in, optional, could go at the end)
vi. The list of figures (not keyed in, optional, could go at the end)
vii. The list of tables (not keyed in, optional, could go at the end)

e. The basic document components
f. Optional acknowledgments
g. Appendices (may contain basic document components).
h. The back material (may contain basic document components).
   i. Bibliography, list of references
      1) Bibliography item
         - Journal information (if the reference is to an article):
           — Author(s) (mandatory)
           — Title (mandatory)
           — Journal name or abbreviation (mandatory)
           — Book or conference proceedings (mandatory)
           — Year of publication (mandatory)
           — Editor of book/proceedings
           — Volume
           — Number in volume of an issue of a journal
           — Page numbers
           — Month of publication
           — Publication note
         - Book information (if the reference is to a book):
           — Author name (mandatory)
           — Book Title (mandatory)

- The Publisher's name (mandatory)
- Year (mandatory)
- Chapter title
- Page numbers
- Title of multi-volume work
- Editor of book
- City (and optionally the country) of publication
- Incidental note

• Report or manual information:
- Author (mandatory, not for manuals)
- Title (mandatory)
- Organisation (mandatory)
- Year of publication (mandatory)
- Chapter title
- Page numbers
- City
- Month
- Reference number
- Note

• PhD Thesis:
- Author (mandatory)
- Title (mandatory)
- University (mandatory)
- Year (mandatory)
- Chapter title
- Page numbers
- City
- Note

ii. Indexes

The indentation reflects the position of the element within the document.

## *D.5 Scientific Articles*

This document type has exactly the same elements as the standard document type, but should be used for preprints. An attribute on the ARTICLE (the highest element in the structure) tag should indicate the language (default English) and which type of article is required. The default will be the EP preprint format. In the case of EP, an attribute for the experiment code will be present. This code will not appear on the document, but will permit storing of the preprint in a special database.

1. The Article. Attributes for indicating language, (experiment) code, security, status and version number. Graphics, mathematics and index entries may appear anywhere in the document.
   a. Publication information (e.g. Nucl. Phys. B166 (1980) 256)
   b. Copyright notice
   c. Keyword list
   d. Front material
      i. The title page
         1) One or more title lines. The possibility to indicate a short title.
         2) The date. Possibility to indicate 'received', 'revised' or 'published'.
         3) One or more reference numbers

4) A general note. Also for updates on earlier versions of the same paper.
5) Conference information.
   - Conference code.
   - Conference title.
   - Conference date.
   - Conference place.
   - Conference prefix: "Presented at:"
6) Journal information
   - Title of the journal.
   - Publication prefix: "Submitted to:"
7) Book/proceedings information.
   - Title of the book/proceedings.
   - The name of the editor of the book/proceedings.
   - The name of the publisher.
   - The place of publication.
   - The date of the publication.
8) A list of authors
   a) A name per author (one mandatory)
      i) Initials per author
      ii) Surname per author
   b) A note per author ("Now at:...")
   c) A line per author
9) A list of addresses
   a) Name of the department
   b) Name of the institute (one mandatory)
   c) Institute qualifier
   d) City
   e) State/County
   f) Country
   g) Postalcode
10) Collaboration name

ii. The abstract (may contain basic document components)
iii. The preface (may contain basic document components)
iv. Acknowledgments (may contain basic document components)
v. The table of contents (not keyed in, optional, could go at the end)
vi. The list of figures (not keyed in, optional, could go at the end)
vii. The list of tables (not keyed in, optional, could go at the end)

e. The basic document components
f. Optional acknowledgments
g. Appendices (may contain basic document components).
h. The back material (may contain basic document components).
   i. Bibliography, list of references
      1) Bibliography item
         - Journal information (if the reference is to an article):
           — Author(s) (mandatory)
           — Title (mandatory)
           — Journal name or abbreviation (mandatory)
           — Book or conference proceedings (mandatory)
           — Year of publication (mandatory)
           — Editor of book/proceedings
           — Volume
           — Number in volume of an issue of a journal
           — Page numbers

- — Month of publication
- — Publication note

- Book information (if the reference is to a book):
  - — Author name (mandatory)
  - — Book Title (mandatory)
  - — The Publisher's name (mandatory)
  - — Year (mandatory)
  - — Chapter title
  - — Page numbers
  - — Title of multi-volume work
  - — Editor of book
  - — City (and optionally the country) of publication
  - — Incidental note
- Report or manual information:
  - — Author (mandatory, not for manuals)
  - — Title (mandatory)
  - — Organisation (mandatory)
  - — Year of publication (mandatory)
  - — Chapter title
  - — Page numbers
  - — City
  - — Month
  - — Reference number
  - — Note
- PhD Thesis:
  - — Author (mandatory)
  - — Title (mandatory)
  - — University (mandatory)
  - — Year (mandatory)
  - — Chapter title
  - — Page numbers
  - — City
  - — Note

ii. Indexes

The indentation reflects the position of the element within the document.

## *D.6 A newsletter layout*

This is a layout that could be like the Computer Newsletter, the CSE Newsletter or the Mini and Micro Computer Newsletter.

1. Front material
   a. First Title page with CERN logo and "CERN" text string
      i. Title lines on the title page
         - Index entries
         - Inline mathematical formulae
         - Character data
      ii. Edition number
      iii. A date
      iv. Graphics
   b. Second Title page

i. Same title as on first title page (formatting can be different).
ii. A list of Editors :
- A name per editor (index entries, character data)
- Institute/address per editor (index entries, character data)
- Notes per editor (index entries, character data)

This list might be generalised to a more general section of reference information.(Where people can find the services provided, who provides them etc.)

c. The table of contents (not keyed in, optional, could go at the end) together with the names of the people contributing. Part of a layout decision might be to put the table of contents on the front page.
d. The list of figures (not keyed in, optional, could go at the end)
e. The list of tables (not keyed in, optional, could go at the end)

2. The basic document components
3. Appendices (may contain basic document components)
4. The back material (may contain basic document components)
   a. Endnotes (optional)
   b. Indexes
   c. Bibliography, references

The indentation reflects the position of the element within the document.

### *D.7 Transparencies*

This layout should contain the same elements as the standard document type; it will not be made available as a separate document type, but rather through an attribute on the highest level tag.

### *D.8 Yellow Reports*

The structural elements are the same as for preprints, with the addition of a copyright notice after the list of authors and before the abstract. However, a completely general yellow report layout should also cater for a conference proceedings layout containing a collection of papers. For the moment, it has been decided **not** to offer a completely general yellow report DTD.

### *D.9 The weekly bulletin*

A completely general DTD for the weekly bulletin could be written and indeed would be useful, as this would enable certain important articles (communications of the Director General) to be stored on-line.

### *D.10 A standard seminar notice*

This is a one-sheet seminar announcement, which could be included into the VM NEWS system or into the weekly bulletin. An attribute is present on the GDOC tag indicating what type of seminar (EP, Theory, Computer, DD, LEP, Technical Presentation, Specific experments etc.):

1. A general seminar notice heading. An attribute indicates the type (Computer, EP, DD etc.)
2. Title. The title of the seminar
3. Speaker. The name of the speaker
4. Date. The date of the seminar

5. Place. The place of the seminar
6. Coffee/tea. Coffee/tea will be served at ....
7. Abstract. The abstract of the seminar
8. About the speaker. A short cv of the speaker

## *D.11 A standard FAX cover page*

This page should be generated independently for any document which is intended for FAXing. Logos belonging to experiments may be added via an attribute (default = CERN logo).

1. From
   a. Name of person sending the FAX
   b. The group of the person sending the FAX
   c. The division of the person sending the FAX
   d. The telephone of the person sending the FAX
   e. The electronic mail identifier of the person sending the FAX
   f. The CERN FAX number of the person sending the FAX
2. To
   a. The name and the address of the person receiving the FAX
   b. The FAX number of the person receiving the FAX
3. The date the FAX was sent
4. The number of pages excluding the cover page

## *D.12 Basic document components*

These may appear in all document types where 'basic document components' are mentioned.

1. Level 0 to 6 headings (parts, chapters, sections...)
   a. Paragraphs
   b. Index term entries
   c. Citations
   d. Short quotations
   e. Long quotations
   f. Footnotes
   g. Endnotes
   h. Figures
      i. Figure Captions
      ii. Figure description
   i. Tables
      i. Table Captions
      ii. Table Descriptions
      iii. Single Cell material
      iv. Rows
      v. Columns
   j. Spreadsheets
   k. Highlighted phrases
   l. Examples
   m. Boxes
   n. Ordered lists
   o. Unordered lists
   p. Simple lists

q. List items
r. List items with headings
s. Definition lists
    i. Definition Terms
    ii. Definition Descriptions
    iii. Glossary lists
    iv. Glossary terms
    v. Glossary descriptions
t. List paragraphs
u. References to all text elements
v. Subroutine listings
w. Graphics
x. Mathematical formulae
y. Inline formulae
z. Display mode formulae

## References


1. J. Ferguson, 'Management Information Systems at CERN', *Management Board paper*, January 1986.

2. E. van Herwijnen, 'Future Office Systems Requirements', *DD Report*, November 1988.

3. ISO, 'The Standard Generalised Markup Language', *ISO 8879*, Geneva 1986.

4. ISO, 'The Office Document Architecture and Interchange Format', *ISO DIS 8613*, Geneva 1987.

5. CERN has installed TEX which was obtained from the standard distribution sites (Maria Code for VM and Kellerman and Smith for VMS). Drivers for the IBM 3812 laser printer and the LN03 laserprinter are installed.

6. Adobe, 'PostScript Language Manual', *Adobe Systems Incorporated*, Palo Alto 1984.

7. M. Weinstein, 'Everything you wanted to know about PHYZZX but didn't know to ask', *SLAC-TN-84-7*, October 1984.

8. Joost G. Kircz and Jan Bleeker, 'The use of relational databases for electronic and conventional scientific publishing', *Journal of Information Science 13*,(1987) 75-89.

9. T. Ericson, Private Communication.

10. IBM, 'Publication Systems PostScript Interpreter', *IBM Product N$^0$ 5688-104*, October 1988.

11. L. van Dam and E. van Loenen, 'An SGML application interface',*CERN DD Report*, June 1989.

12. Jurgen de Jonghe, 'Text Processing at CERN, Part II: The DTDs for CERNs SGML implementation', *CERN DD Report*, to be published.

13. Microsoft, 'Using Microsoft Word, Version 4.0', *Microsoft Corporation*, 1987.

14. MIS, 'MIS Reference Book', *CERN MIS Unit*, sections 1.7.1 and 1.7.2, Geneva April 1988.

15. P. T. Warren, 'SGML and style sheets: the implications for electronic document preparation', *University Computing*, 1987, **9**, 81-86.

16. Dee Cork and Jurgen de Jonghe, 'A user guide for MS Word to SGML conversion, (PC version)' *CERN DD Report*, CERN June 1989.

17. Microsoft, 'Rich Text Format specification', *Microsoft*, 1988.

18. Michael Spivak, 'PC TEXManual, version 2.1', *Personal TEX Inc.*, 1988.

19. Paul Hofman, 'Microsoft Word for the Macintosh made easy, Version 3.0', *McGraw-Hill*, Berkeley 1987.

20. SoftQuad, 'Author/Editor™ Advance version Users Guide', *Softquad Inc.*, Toronto June 1988.

21. Mathtype, 'The Mathematical Equation Editor for Macintosh, Version 1.53', *Design Science, Inc.*, Long Beach 1987.

22. Two acceptable Macintosh $T_EX$ systems are TEXTURES and MACTEX.

23. A. Scheller and C. Smith, 'DAPHNE, Document Application Processing in a Heterogeneous Network Environment, Version 2.0', *Benutzerhandbuch*, DFN-Bericht Nr 41, Berlin July 1986.

24. Jos Warmer and Sylvia van Egmond, 'The Implementation of the Amsterdam SGML Parser', Amsterdam 1988.

25. Waterloo, 'Waterloo SCRIPT Reference Manual for Waterloo SCRIPT, Version 86.1', *Department of Computing Services, University of Waterloo*, Waterloo, February 1987.

26. IBM, 'IBM SGML Translator DCF Edition', *IBM Product N° 5684-025*, September 1988.

27. IBM, 'IBM Document Composition Facility Generalized Markup Language Starter Set Reference, Release 3.2', *IBM Manual SH20-9187-05*, Boulder 1988.

28. IBM, 'Live Parsing Editor Reference', *IBM Product N°: SU59-7008*, Winchester June 1987.

29. Computing at CERN in the 1990s, 'Final Report of the working group for experiments, section 10.4 Interactive Data Analysis', *CERN Internal report*, 21 October 1988.

30. Interleaf, 'Technical Publishing System, version 4.0', *Interleaf, Inc.*, Boston 1987.

31. IBM, 'Browsemaster Licensed Program Specifications', *IBM Manual N° GH23-6090*, New York 1988.

32. FTP, 'PC/TCP, Network Software for DOS', *FTP Software Inc.*, Boston 1987.

33. Peter di Camillo, 'TN3270 for the Macintosh Reference Guide, Version 2.0', *Brown University*, July 1987.

34. ISO, 'EDI for Administration Commerce and Transport', *ISO XXXX*, Geneva 1987. The subcommittee responsible for this standard will be alerted by ISO/IEC JTC 1/SC 18 of the suitability of SGML for TDI and EDI. See also the SGML Users' Group Newsletter, N° 8, August 1988, p.4.

35. ISO, 'Operational Model for Text Description and Processing Languages', *ISO/TC97/SC18/WG8 N 484*, August 1987.

36. D. Chamberlin et. al., 'QUILL: An Extensible System for Editing Documents of Mixed Type', *Proceedings of the 21st Hawaii International Conference on System Sciences*, The Computer Society of the IEEE, Washington, January 1988.

37. DEC, 'Guide to VAX Language-Sensitive Editor and VAX Source Code Analyser, Version 2.1', *DEC Order N°: AI-FY24B-TK*, July 1987.

38. Sobemap, 'The Mark-It Manual, version 2.0', *Sobemap S.A.*, Brussels June 1988.

39. SGML Newsletter N$^0$ 10, page 11, November 1988.

40. Lawrence Ellison, 'Oracle Overview and Introduction to SQL',*Oracle Corporation*, May 1985.

41. E. van Herwijnen, L. van Dam, M. Nijdam, 'Storing and retrieving SGML documents', *CERN DD Report*, to be published.

42. ISO, 'SGML Document Interchange Format', *ISO 9069*, Geneva, September 1988.