SGML (, TEX and ...)

C.G. van der Laan
Rekencentrum RUG
Landleven 1, 9700 AV
Groningen, The Netherlands
050-633374/8080

Abstract

An introductory review is given of what SGML (and DTDs), TEX (and formats), and their relation, is all about. Coupling SGML to TEX is considered via interfacing and transformation. Transformations of the tags of the *basic* as well as the *complete* SGML marked up compuscript are dealt with in detail for the examples: a letter, bridge lay-out, some mathematics and a simple table. At the end some guidelines are provided in order to decide when SGML, TEX, or both, might be beneficial, along with a conclusion. It is a 3-in-1 paper: (1) what SGML and TEX is all about, aimed at novices; (2) examples of marked up copy in SGML and (La)TEX and the coupling issues, for those already familiar with SGML and TEX, but like to be informed about transformation issues, when mathematics or tables are part of the compuscript; (3) finally, a literature compilation, for those who consider bibliographies at hand useful.

What is a Document?

The Association of American Publishers (AAP) *Reference Manual on Electronic Manuscript Preparation and Markup* defines a document as:

> A document is an organized collection of smaller pieces of text (such as chapters) and images (such as figures) that are called elements. The elements in a document have a relationship to each other which gives the document a definite organization called document structure.

Lifecycle-phases of documents. Compuscript preparation requires that additional information be interspersed within the text to aid any subsequent processing. That information, called markup, is usually specific to a particular publisher, system or printing device. A universal standard method of marking up electronic compuscripts, however, offers many advantages.

The *complete Lifecycle* of a document can be thought of as:

- preparation,
- distribution,
- reading,
- storing (Paper? Electronically? Optically?),
- other usage, reuse?[1]

[1] Publishers like reuse, authors rework!

Standard Generalized Markup Language (SGML) supports the complete Lifecycle, where *future* usage of the document is not necessarily restricted to printing. SGML must be complemented, however, by generally accepted Document Type Definitions (DTDs). The Association of American Publishers [AAP, 1987 and 1989] and the British National Bibliography [Smith, 1987] have provided some DTDs. In order to serve the primary aim of publishing, the coupling to formatters must be supported too.

TEX supports formatting and electronic document exchange.

What is SGML?

SGML provides a language to describe documents and has made it possible to achieve two goals:

1) establish a standard means of identifying and tagging parts of an electronic compuscript so that computers can differentiate between these parts; and
2) provide some logical ways of representing special characters, symbols and tabular material, using only the ASCII character set usually found on standard keyboards.

SGML is defined in ISO8879 [1986]. Some introductions to SGML are: Barron [1989], gentle SGML in Sperberg-McQueen & Burnard [1990], and the books Bryan [1988] and van Herwijnen [1990]. Examples are provided in among others ISO/TR9573 [1988].

Purpose. The purpose of SGML is to facilitate *information* exchange, and reusability (in other contexts, even yet unknown contexts),

— *Then and There* [2] —

via a description *language*, where information is packed in compuscripts, containing, text, graphics, formulas, tables, etc.

Meta Language. SGML is a *meta* language, which can be used to define an arbitrary number of markup languages in a standardized way. This means, for any class of documents, markup rules can be prescribed by SGML, yielding a *language*, the Document Type Definition, for that class. The parser checks compliance of the marked up copy to the DTD.

Standard.

Formerly: No consensus on markup codes (WordPerfect, Wordstar, MacWrite, ...; Scribe, troff, TEX, LATEX, ...)

Presently: SGML ISO standard

Standard $\stackrel{\text{def}}{=}$ It can be used to define an arbitrary number of markup languages in a *standardized* way.

Entails:

General applicability,
Extended lifetime,
Improved reusability,
Enhanced exchange possibilities.

Generalized.

Formerly: (Typeset) *marks* for *specific* here-and-now printers, via *direct* markup.

Presently: Marks are *generic.*[3] This is done with procedural markup. Macro calls are inserted as markup tags, where the implementation of the macros (the format or style file) represents the style, accounts for the fonts, etc. The printer hardware is shielded by intermediate languages. Intermediate language copy is printed via drivers. Change of style needs another style file, no modification of tagged copy is necessary. Change of printer hardware needs another driver, no modification of tagged copy nor modification of format file!

[2] The essential issues of portability: portability with respect to time (then) and place (there).

[3] As opposed to specific print/plot/photo typeset hardware.

Generalized $\stackrel{\text{def}}{=}$ Abstraction from the specific (printing) to the general (other usage), by emphasizing the *structure* of a document and to specify intent without regard for appearance.

Markup.

Formerly: (Typeset) *marks* in the margin (Marks are bound to a version; no "data-integrity")

Presently: Marks are integrated along with copy (Data-integrity of markup code is preserved.)

Markup $\stackrel{\text{def}}{=}$ Term used to describe codes added to the electronically prepared document.

Author's point of view. Authors have to markup their copy with

1) Awareness of the DTD which applies to the document; either the DTD must be understood or templates must be available;
2) Knowledge of which (begin) tags to use where and how;
3) Knowledge of ranking, attribute use, tag minimization;
4) Knowledge about how to create entity references.

These aspects are treated in author's guidelines. The above can be alleviated by providing an SGML *environment*, or better, a document preparation environment, supported by menus and templates with prompts. Thus authors can concentrate on structure and content by using a standard (generic) markup language, or a sufficiently advanced document workbench, leaving formatting issues to publishers, or software vendors. Because of this "separation of concerns" the author's task is simplified.

Publisher's point of view. Publishers make use of sufficiently accepted DTDs. They provide authors with guidelines and proofing tools. DTD writing requires knowledge of the various types of markup, such as presentational, direct, procedural and descriptive markup.

Example markups.

No markup. In order to remind the unpleasant look of documents with just words, we start with a no markup example.

```
TeX A system for formatting text
TeX and the accompanying macro
package LaTeX provide powerful means ...
```

Presentational markup. Documents with tabs, indentations, and in general positional control make use of what is called presentational[4] markup, in order to convey the meaning.

[4] Some editors prefer submission of this form for simple text!

```
TeX:
A system for formatting text.

  TeX and its accompanying macro
package LaTeX provide
powerful means of formatting
text to be output on either
  - a simple matrix printer,
  - a laser printer or
  - a photo typesetter.
```

Presentational markup is functional with poetry, such as Alice's mousetail as mentioned by Malcolm Clark [1989] or DEK's favourite poem of Piet Hein [*The Errors of TEX*, 1989], where the words are arranged along an ellipse.

Direct markup. When specific print instructions are included, we get direct markup:

```
@T:                             []
A system for formatting text.[]
                                [I]
@T and its accompanying macro
package @LT provide
powerful means of formatting
text to be output
on either                       [I]
 - a simple matrix printer,     [I]
 - a laser printer or           [I]
 - a photo typesetter.
```

`[I]` is a print instruction indicating to go to the next line and *indent*; @<name> stands for a process with a special formatting effect.

Procedural (LATEX) markup. A markup command, where the implementation of the command contains print instructions, is considered a procedural markup command; when the printer is changed the implementation has to be changed too, not the marked up copy. LATEX markup of the example reads

```
\subsection*{\TeX}
A system for formatting text.
\par
\TeX\ and its accompanying macro
package \LaTeX\ provide
powerful means of formatting text
to be output on either
\begin{itemize}
\item simple matrix printer,
\item a laser printer or
\item a photo typesetter.
\end{itemize}
```

Descriptive (SGML) markup. Descriptive markup goes even further and uses markup which describes the structure and intent of the various parts of the document:

```
<h>&TeX;
<p>A system for formatting text.
<p>&TeX; and its accompanying macro
package &LaTeX; provide
powerful means of formatting
text to be output on either
<li>
<it>simple matrix printer,
<it>a laser printer or
<it>a phototypesetter.
</li>
```

Formatting information with SGML. It is possible to convey formatting information via SGML. This is done with element attributes or with Processing Instructions. Other symbols than those in the ASCII character set are often denoted by an entity reference to the font containing those symbols, with appropriate loading of the font elsewhere. With respect to attributes, one can think of specifying open space in order to include illustrations from other (electronic) sources. Also, indication of a representation choice is possible if properly accounted for in the DTD. Consider for example the representation of labels of list items: alphabetical or roman/arabic numeral.

```
<li number=alpha>
<it>a simple printer,
<it>a laser printer or
<it>a photo typesetter.
</li>
```

It is possible in SGML to include document parts which already contain formatting information. The parser must be told to lay back. For a notation to be allowed it must be declared via e.g.

```
<!NOTATION TeX SYSTEM>
```

for TEX formatted copy. Appropriate entity and attribute specifications are also needed in the DTD. For an author the equation formatting with the type attribute (value=TEX) has to be supplied as

```
<eqn type=tex>
  $$X\cap(A\cup B)=(X\cup A)\cap(X\cup B)$$
</eqn>
```

Processing Instructions (PIs) can be used to tell the local system how it should process data contained within a document. For example, SETM typography markup instructions:

```
<p><?[s24][sec][rm]>T<?[pri][rm]his>
          paragraph ...
```

In this case the SETM instructions are in brackets, preceded by the PI open delimeter `<?`. The meaning of this instruction is to treat "**T**" as "**24**pt" initial letter to be set using the **rom**an version of the face currently defined in the **sec**ondary type family.

Availability. As mentioned by Herwijnen [1990], Sobemap, The Publisher and IBM's SGMLDCF, are some SGML systems that are already available.

Support. Support for SGML is done by the companies, as part of automation projects. There also exists a Dutch chapter of the SGML Users Group.[5]

Courses. Courses are provided by private companies, and the National Normalization Institutes.

What SGML is not! SGML is not:

- WYSIWYG (pronounced wĭsēwĭg, and stands for what you see is what you get),
- A formatter, certainly not a standard formatter.

What is TEX?

TEX stands for $\tau\varepsilon\chi$, the first three letters of the Greek word for *technique*, which also means art. TEX is a *machine independent* formatting language designed by Don Knuth, [*The TEXbook*, 1990 (Version 3.0)]. Michael Doob gives an easy start to TEX in *A Gentle Introduction to TEX* [1989]. A systematic treatment of the commands is given by Abrahams [1990]. There also is an introduction in French by Seroul [1989] and various German introductions: Appelt [1988], Schwarz [1989]. Von Bechtolsheim [1990, in English] is impressive. LATEX, by Leslie Lamport [1985], is a macro collection for simplified use of TEX. LATEX uses *procedural* markup. Buerger [1990] gives a LATEX introduction. In German there are Kopka [1989] and Wonneberger [1987]. Bruin [1989] gives a Dutch introduction to LATEX.

Purpose. The stated purpose of TEX is "making beautiful books."

Processing (La)TEX. LATEX is processed in three steps: edit the copy, format the copy to create a `dvi` file, and print the resultant `dvi` file. A diagram of this looks like:

$$\text{copy} \xrightarrow{editor} \texttt{ASCII} \xrightarrow{(La)TeX} \text{dvi-file} \xrightarrow{driver} \text{results.}$$

The more steps to the process, the more cumbersome is correction handling because of larger production *loops*. This is the case when the use of TEX is combined with SGML markup. The SGML parsing and linking extends the loop.

[5] SGML-Holland secretary: D. van Wijnen, Wolters Kluwer. P.O. Box 989, 3300AZ Dordrecht. 078-334933; e-mail: surf003@kub.nl.
SGML User's Group secretary: S. G. Downie, Softquad Inc, 720 Spadina Avenue, Toronto, Ontario M5S 2T9, Canada.

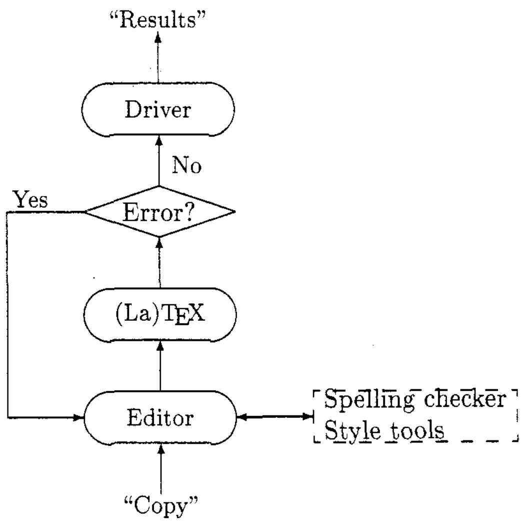

Figure 1: Correction cyclus

Availability. TEX is available on many computers under various operating systems with a variety of drivers for previewing (such as VDU), printing, and photo typesetting. Documents written in TEX or LATEX can be ported easily. Exchanging documents via e-mail is also generally possible except for the incorporated graphics. When graphics are part of the document, TEX can be combined with (encapsulated) Postscript, which is used within the TEX community and elsewhere. Portability is restricted to places with Postscript printers of course. TEX is in the *public domain*. Drivers are generally not. They do, however, generally have value added by the companies you buy the driver from. See the ads in any TUGboat. Moreover, TEX systems can make

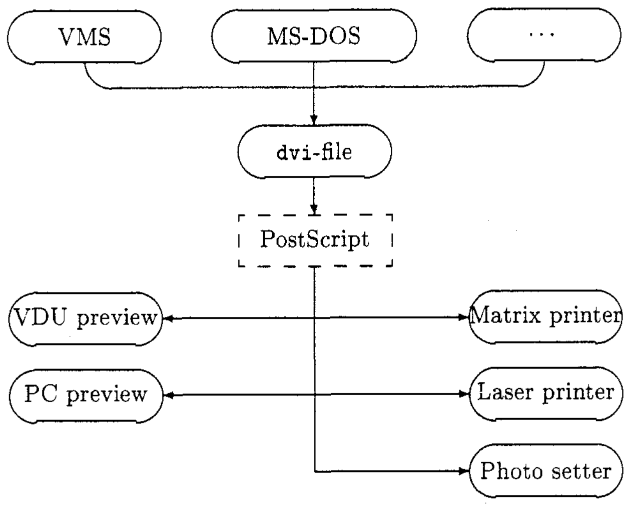

Figure 2: (La)TEX's use

use of fonts from various sources, such as Adobe's PostScript fonts and, of course, Metafont.

Support. Support is organized by the various users groups. Software, style files, macros etc. are distributed (via e-mail, ftp or floppy disks) by the TEX Users Group (TUG)[6]; in the Netherlands it is distributed by NTG[7]; in France by GUTenberg; in Germany by DANTE; in the United Kingdom by ukTEXug; in the Nordic countries by "Nordic TuG". There is also the recent TUGlib (analogous to NETlib from the numerical mathematicians) service in Utah: e-mail fileserver, literature surveys (a.o. what has appeared in TUGboat), and the who-is database.

Courses. Courses are organized by TUG and other TEX user groups, especially in conjunction with their main meetings.

Relationship: DTDs, SGML, TEX, formats and ...

The relationship of TEX, SGML and other applications is illustrated in the diagram below. An integrated[8] implementation is Arbortext's "The Publisher," which has AAP's DTDs built-in, and runs on a.o. SUN hardware. Note that the backarrows denote some of the work in progress by Elsevier Science Publisher, Bleeker [1989], Poppelier [1990].

SGML ór TEX sufficient? NO, needed are format files and DTDs as well! If a compuscript is printed with TEX for personal use only, there is nothing to worry about. When no reuse of a document is in sight, but remote publishing and electronic exchange are considered, it pays to use standard format/style files — which reflect the lay-out of the document type — along with TEX. When reuse or abstract structuring are being used, standard DTDs will be crucial.[9]

Moreover friendly user-interfaces are needed. Grootenhuis (priv. comm.) provided on top of the SGML system he used, the possibility to input letter copy in a natural way with tags hidden: awareness of the DTD used is not necessary, nor does the user see or have to type in any tags. (Of course a sample of how the letter looks is needed.) This kind of system needs a high degree of foolproof-ness: the spaces etc. have *side effects* and when errors are made the (SGML) messages are quite confusing. Another possibility is to prompt the user for the required copy, while the tags are provided by the system.

[6] Editorial and TUG address: TEX Users Group, P.O. Box 9506, Providence RI 02940, USA. email: TUGboat@Math.AMS.com.

[7] NTG: Nederlandstalige TEX Gebruikersgroep. Secretary: G.J.H. van Nes, Postbus 394, 1740AJ, Schagen, 02246-4185; e-mail: vannes@ecn.nl, ntg@hearn.

[8] Ikons user interface, SGML layer, TEX layer, Postscript handling (optionally); with SGML, TEX and dvi files as intermediate results

[9] The flexibility and open-endedness of SGML are strong points, everybody can write or modify DTDs; this is also a weak point because of incongruent DTDs.

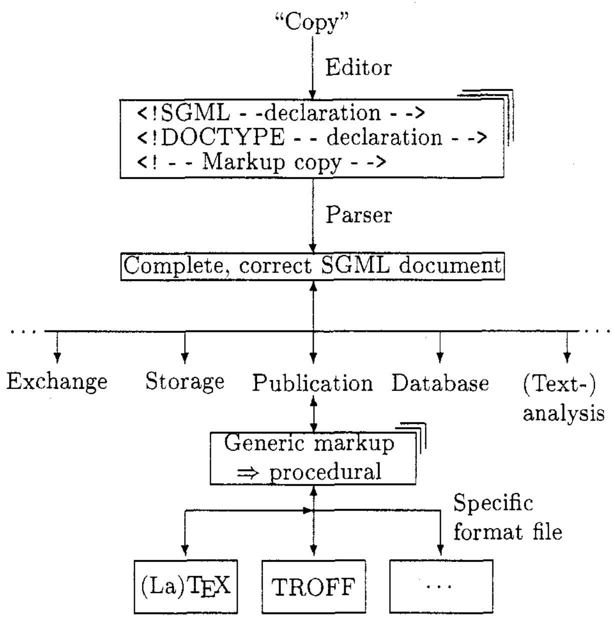

Figure 3: Relation SGML and (La)TEX

Interfacing vs. transformation. Interfacing copy marked up by any formatter to SGML is possible in SGML via the NOTATION mechanism. Of course, it has to be incorporated into the DTD and appropriately implemented: the parser should lay back and leave the formatting of the TEX marked up copy to TEX.
Transformation SGML into TEX is different. Using TEX as a back-end formatter to SGML can be done. It is supported by the link mechanism of SGML. Needed is at least a table of corresponding notations in order to substitute the markup tags from SGML into TEX. The other way round has to be done by a separate program. In the sequel we will study example document elements marked up by SGML and (La)TEX; transformation issues will be addressed as well.

Examples

Simple text. As an example let us take the simple text given earlier. The (basic) SGML and LaTeX markups have been given in previous sections. Coupling comes down to a change of representation, except for the omitted endtags. This direct approach needs the substitutions:

SGML	⇒ TeX
`<h>`	`\section{`
`<p>(first)`	`}`
`<p>`	`\par or blank line`
`<li>`	`\begin{itemize}`
`</li>`	`\end{itemize}`
`<it>`	`\item`
`&TeX;`	`\TeX␣`
`&LaTeX;`	`\LaTeX␣`

This suggests systematic coupling of all entities and tags to equivalents in LaTeX. The explicit endtags are more natural to handle than the omitted ones. The handling of the first and following occurences of `<p>` have to account also for respectively finishing the heading `</h>` and ending a paragraph `</p>`, which have been omitted.

Letter. A typical letter has the general aspects:

- Background
  Heading (Logo, address, phone, ...)
  Footer (numbering, ...)
- Context (running heads next pages, ...)
- Reference
- Your reference
- Date
- Addressee (name, company, address, zip code)
- Beginning (Dear...)
- Contents
- End matter (Salutation, name, position)
- Additions (PS, enclosure, cc)

The above aspects are generally ordered in a tree for detailing the hierarchical structure and writing the DTD from.[10] To overcome the optional and repetition notational difficulties in the structure tree for document parts, signs are added to the knots: + for repetition, and * for an optional element. (This mechanism accounts also for 'elements in arbitrary

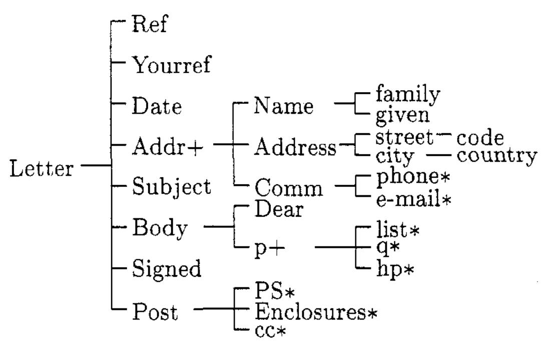

Figure 4: Hierarchical letter structure

order.') Attributes are not reflected in such a tree, neither are included and excluded elements. It is not clear to me why the tree structure is chosen instead of the syntax flowchart notation as used, e.g., along with the definition of the Pascal programming language. From such a tree a DTD is generally written. Only the bridge example DTD is used in this article, because the rigorous DTDs, how interesting and instructive they might be, would lead us too far away into SGML details.

SGML markup. The SGML markup for a typical letter might look something like:

```
<!DOCTYPE letter PUBLIC
 -- DTD to be used  --
  "-//NTG//DTD Letter//EN">
<letter -- start-tag -->
<ref> CGL/Ba/B89-007
<yourref> MC/L1/L89-001
<date> 4 august 1989
<address> Malcolm Clark, ICRF
<email>
     malcolm@icrf.ac.uk
</email>
<dear>Malcolm
<p> Thank you very much ...
...
<p> Some details about the course ...
...
<signed name=CGL>
</letter -- end-tag -->
```

LaTeX specification. The same letter using LaTeX markup might look like:

```
\documentstyle[12pt]{letter}
  \address{% return address
          C. G. van der Laan   \\
          \ldots}
  \signature{Kees}
\begin{document}
```

[10] A difficulty is what to exclude from the structure and to consider as a formatting issue. In the letter example I have considered the header (with logo) and the headerlines on the pages 2 etc., and the footers as formatting issues. They could have been included in the DTD as included elements; this cannot be represented in the structured tree.

```
\begin{letter}{% address
          Malcolm Clark       \\
          \dots}
% no ref or your ref
% date is handled automatically
\opening{Dear Malcolm}
\par
Thank you very much \ldots
\begin{quote}
$\vdots$
\end{quote}
Some details about the course \ldots
\begin{quote}
$\vdots$
\end{quote}
\closing{Best regards}% Handles signature
%ps, cc, enclosure all possible
\end{letter}
\end{document}
```

Letter result. Because a sample LaTeX letter could not be processed simultaneously in this paper (I don't have access to Postscript facilities in order to include the separately produced result), the printed letter has been omitted.

Transformation. What comes to mind when looking at both representations of marked up copy is the difference in the sequence order of tagged items in SGML and LaTeX. With *complete* marked up SGML copy to be transformed into procedural TeX, there is no problem: in TeX macros, strings can be stored for later usage. This will be shown along with the tabular example in the section 'transformation revisited.'

Bridge card deal. In TUGboat 11#2 I have described typesetting bridge using TeX.

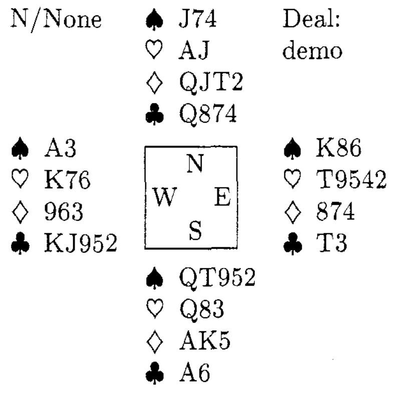

Figure 5: Bridge deal

SGML markup. The SGML markup for Figure 5 above might look like:

```
<deal><vuln>N/None
      <comm>Deal: demo
<hand n><spades>J74
        <hearts>AJ
        <diams> QJT2
        <clubs> Q874
<hand e><spades>K86
        <hearts>T9542
        <diams> 874
        <clubs> T3
<hand s><spades>QT952
        <hearts>Q83
        <diams> AK5
        <clubs> A6
<hand w><spades>A3
        <hearts>K76
        <diams> 963
        <clubs> KJ952
</deal>
```

(La)TeX specification. The procedural TeX markup for Figure 5 might look like[11] :

```
\crdima{N/None}{\vtop{\hbox{Deal:}
                      \hbox{demo}}}%
  {\hand{J74}{AJ}{QJT2}{Q874}}%N
  {\hand{K86}{T9542}{874}{T3}}%E
  {\hand{QT952}{Q83}{AK5}{A6}}%S
  {\hand{A3}{K76}{963}{KJ952}}%W
```

Transformation. The transformation comes down to

SGML	⇒ TeX
`<deal>`	`\crdima`
`<vuln>N/None`	`{N/None}`
`<comm>DEAL: demo`	a suitable `\vtop`
`<hand x>`	`{\hand`

and all the cards per colour surrounded by curly braces, with an extra "}" after the clubs. Although once again a simple[12] example, the translation table is not natural. Note. This is a bottom-up example: descriptive markup TeX macros were already available. Starting from descriptive TeX marked up copy is simple, and a candidate for table-driven substitution, apart from some exceptions like the suitable \vbox here.

TeX macros. Figure 5 is created by using the TeX macros:

```
\def\hand#1#2#3#4{%
%Example: \hand{AKJ765}{AK9}{--}{T983}
\vtop{\hbox{\strut\s\enspace#1}
```

[11] \minipage is not used because of the more general \vtop command, which can be used as well in LaTeX as in TeX.

[12] If one ever can call nested tables simple.

```
\hbox{\strut\h\enspace#2}
\hbox{\strut\d\enspace#3}
\hbox{\strut\c\enspace#4}}%end \vtop
}%end \hand
%
\def\crdima#1#2#3#4#5#6{%
%purpose: layout bridge hand
%#1 left upper·text
%#2 right upper text
%#3, #4, #5, #6: N, E, S, W hands
\vbox{\halign{                 &##\quad\cr
             #1&           #3&      #2\cr
 $\vcenter{#6}$&$\vcenter{\copy\NESW}$&
                          $\vcenter{#4}$\cr
               &           #5&        \cr
              }%end \halign
      }%end \vbox
}%end \crdima
%
\def\NESWfig{%
\vbox{\font\small=cmr9
\def\str{\vrule height2.2ex%
    depth.75ex width 0pt}
\offinterlineskip\tabskip0pt\hrule
\halign{\vrule\hskip2pt\relax
##\hfil\tabskip3pt&  \str\hfil##\hfil&
##\hskip2pt\relax\hfil\vrule
                         \tabskip0pt\cr
  &      \hbox to 0pt{\hss\N\hss}&  \cr
\W&                    \phantom{N}&\E\cr
  &  \str\hbox to 0pt{\hss\S\hss}&  \cr
       }%end \halign
\hrule}%end \vbox
}% end \NESWfig
\setbox\NESW\hbox{\NESWfig}
```

SGML requirements. The following DTD is needed:

```
<!ENTITY % ISOpub PUBLIC
 "ISO 8879-1986//ENTITIES Publishing//EN">
<!ELEMENT deal - - (vuln, comm?, hand+)>
<!ELEMENT (vuln|comm) - O CDATA>
<!ELEMENT hand - O (spades,
                    hearts, diams, clubs)>
<!ATTLIST hand nesw (n|e|s|w) #REQUIRED>
<!ELEMENT (spades|hearts|diams|clubs)
                     - O CDATA>
```

Note. In the DTD we could have imposed sequence ordering by changing hand+ into (handn, hande, hands, handw). But this requires that all the hands are needed and that is too restrictive.

Some math. The following examples of mathematical formulas are borrowed from the "Mathematical Formulas" report [van der Laan, Coleman, Luyten, 1989]. In this report, SGML and LATEX markup are supplied for formulas from various fields: elementary mathematics, set theory, geometry, functional analysis, calculus (differential equations, special functions, continued fraction), statistics, algebra (tensor calculus), homology (diagrams) and quantum mechanics. A few simple ones are selected here.

LATEX results. The following was formatted with LATEX markup:

$$X \cap (A \cup B) = (X \cup A) \cap (X \cup B)$$

$$x \notin A \not\subset B$$

$$\|a(x+y)\| \leq |a|.(\|x\| + \|y$$

$$\int \frac{1}{\sqrt{1+x^2}}\,dx = \log(1+\sqrt{1+x^2})$$

(Basic) SGML markup. To accomplish this with SGML markup, you might enter:

```
<fd>X&cap;(A&cup;B) =
     (X&cup;A)&cap;(X&cup;B)</fd>

<fd>x&nisin;A&nsub;B</fd>

<fd><fen d>a(x+y)<rp d</fen>&le;
  |a|.(<fen d>x<rp d</fen>
     +<fen d>y<rp d</fen>)
</fd>

<fd><in><opd><fr>1</><rad>1+
  x<sup/2/</rad></fr>dx</in>=
  <rf/log/(1+<rad>1+x<sup/2/</rad>)
</fd>
```

The DTD used is an adapted version of AAP's DTD by D.C. Coleman.

(Direct) TEX markup. LATEX and TEX markup are very similar for these examples:

```
X\cap(A\cup B) =(X\cup A)\cap(X\cup B)

x \notin A \not\subset B

\|a(x+y)\| \leq |a|.(\|x\|+\|y\|)

\int\bfr1</>\sqrt{1+x^2}\efr dx =
                 \log(1+\sqrt{1+x^2})
```

Transformation. To accomplish the SGML to TEX transformation, some general substitutions are needed:

SGML	⇒ TEX
`<fd>`	`\[ or $$`
`</fd>`	`\] or $$`
`<sup/2/`	`^2`
`etc.`	

For the first set theory example the following substitutions are additionally needed

SGML	⇒	TeX
`&cap;`		`\cap␣`
`&cup;`		`\cup␣`

Functional analysis required moreover

SGML	⇒	TeX
`<fen d>`		`|`
`<rp d>`		`|`
`&le;`		`\leq␣`

For the integral the following definition (format) is needed for the fraction, where use is made of `</>` as parameter separator:

```
\def\bfr#1</>#2\efr{{#1\over#2}}
```

Also needed are the substitutions

SGML	⇒	TeX
`<in>`		`\int␣`
`<opd>`		`\relax`
`<fr>`		`\bfr`
`</fr>`		`\efr`
`<rad>`		`\sqrt{`
`</rad>`		`}`
`<rf/log/`		`\log␣`

Note. A translation table is once again not straightforward; unnatural are the SGML difference in norm open and closing, and the fancy use of `</>`, i.e. null endtag for numerator and omitted opening tag for denominator. The short reference for the modulus sign is neat.

(Complete) SGML markup. The sobemap parser yielded the following (visually edited) result for the set theory example:

```
<FD DCN="GEO.FORM">
<FL>
X&cap;<FEN STYLE="S" LP="PAR">
        A&cup;B
      <RP STYLE="S" POST="PAR"></FEN>
=
<FEN STYLE="S" LP="PAR">
   X&cup;A
<RP STYLE="S" POST="PAR"></FEN>
   &cap;
<FEN STYLE="S" LP="PAR">
   X&cup;B
<RP STYLE="S" POST="PAR"></FEN>
</FL>
</FD>
```

This shows that complete SGML is verbose. For example, consider the complete tags for parentheses "(" and ")". Thanks to the short reference mechanism the input can look natural. Coupling of the above to TeX can be done. How to automatically handle attributes in the best way is not yet clear to me. It is not efficient for parentheses, "(" and ")", to be expanded first by the parser into a fence tag with the appropriate attribute value, followed by substitution at the TeX level into "(" and ")" again.

Because matrices are treated similarly to tabular material, we have omitted a matrix example and refer to the next sections, where a table is studied.

AAP's simple table. A simple table is characterized by: simple table entries, one header row, no header subrows, no footer, no intra referencing, and no caption. From the SGML technical point of view no attributes are used. AAP's example simple table [Markup of Tabular Material, 1989], even more simplified, is reproduced below.

Table AAP Job Changes: 1973–1980

	Gain/Loss of Hospitals since 1973	Total No. of CEO Job Changes 1973–80	Survival Rate of CEO's
Texas	+20	—	22%
Maryland	+ 5	42	24%

Source: David Kinzer, "Turnover Of Hospital Chief Executive Officers: A Hospital Association Perspective," *Hospital and health Services Administration* May–June 1982.

Figure 6: AAP's simplified table

(Basic) SGML markup. The SGML markup for Figure 6, with a minor structural adaptation and some layout modifications, reads:

```
<tbl  -- start of table                     -->
<no>Table  AAP
<tt>Job Changes: 1973&ndash;1980
<th   -- heading                            -->
<th>Gain/Loss of Hospitals since 1973
<th>Total No. of CEO Job Changes
    1973&ndash;80
<th>Survival Rate of CEO's
<bdy  -- table body                         -->
@Texas|20||22%
@Maryland|5|42|24%
<ft   -- table foot                         -->
<au>David Kinzer
<atl>Turnover Of Hospital Chief
     Executive Officers: A Hospital
     Association Perspective
<nme>Hospital and health Services
     Administration
<dt>May&ndash;June 1982
</tbl>
```

A DTD for simple tables is not separately provided by AAP; it is incorporated as part of the complex table DTD. The "simple table"-example obeys the following SGML structure description

```
<!ENTITY row    STARTTAG "row"          >
<!ENTITY column STARTTAG "c"            >
<!ELEMENT tbl       - - (hd, bdy, ft)   >
<!ELEMENT hd        - O (no?, tt?, th+) >
<!ELEMENT bdy       - O row+            >
<!ELEMENT row       - O c+              >
<!ELEMENT ft
        - O (au|src|atl|nme|dt)         >
<!ELEMENT (th, c, au, src, atl, nme, dt)
        - O (%t.cs;) -- Character string-->
<!SHORTREF tablemap "@" row
                    "|" column          >
```

Note. Some tags are presumed part of the general DTD, e.g. no, tt.

(Direct) TEX markup. As expected, some formatting commands had to be included in order to reproduce the published results.

The approach has been to supply a template (preamble) line for the layout of the contents proper and to handle the header row as an exception to the template, *manually!* Separators between header, body and footer have to be incorporated as well. A translation table is clearly insufficient. The TEX markup, then, would look like:

```
\newdimen\entrywidth%\entrywidth=<default>
\newdimen\tablewidth
\tablewidth=.5\hsize%default
\def\nl{\par\noindent}
\def\ndash{--}
\def\tablerule{\noalign{\hrule}}
\newdimen\digitwidth\setbox0=\hbox{\rm0}
         \digitwidth=\wd0
%?-command for nonsignificant leading
%          zeroes, Knuth p241
\catcode`?=\active
\def?{\kern\digitwidth}
%\btbl %AAP's simple example with direct
%       TeX markup
\entrywidth=2cm
\tablewidth=4\entrywidth
\vbox{\hsize=\tablewidth
\halign{\hbox to\entrywidth{#\hss}\hfil&&
    \hbox to .5\entrywidth{\hss#}\hfil\cr
%preamble line
   \tablerule\noalign{\vskip1ex}
   \omit{\bf  Table AAP}
        Job Changes: 1973--1980
   \hidewidth\cr
   \tablerule\noalign{\vskip1ex}
   \omit           &
   \omit\vtop{
        \noindent\hsize=\entrywidth
        Gain/Loss\nl
        of Hospitals \nl
        since 1973}&
   \omit\vtop{
        \noindent\hsize=\entrywidth
        Total No.    \nl
        of CEO       \nl
        Job Changes  \nl
        1973--80}  &
   \omit\vtop{
        \noindent\hsize=\entrywidth
        Survival     \nl
        Rate of      \nl
        CEO's}     \cr%end header row
\noalign{\vskip.5ex\hrule\vskip.5ex}
%head-body separation
   Texas    &$+$20&---& 22\%\cr
   Maryland&$+$?5&42 & 24\%\cr
\noalign{\vskip1ex}%body-foot separation
   \noalign{Source: David Kinzer,
   ``Turnover Of Hospital Chief Executive
     Officers:
     A Hospital Association Perspective,''
     {\it Hospital and health Services
     Administration\/} May--June 1982.
    }%end \noalign
}%end \halign
}%end \vbox
```

Transformation revisited.

Complete SGML markup with procedural TEX markup representation can be achieved via the SGML link mechanism, or by any automatic substitution process (programmable editor, preprocessor). Translation into TEX can be done *direct* or via procedural markup. For the mathematical examples given in van der Laan, Coleman [1989], this has been done by Grootenhuis [1990]. He has used the SGML link mechanism for "substitution" and did direct coupling to LATEX. The *direct* coupling approach, without procedural (LATEX) markup, has the disadvantage that change of formatter requires (TEX) source adaptation. In order to abstract from any particular format, we have considered procedural TEX markup as an intermediate phase, in van der Laan, Coleman [1990].

SGML⇒TEX using procedural markup. The "generalized markup ⇒ format" process can be characterized by the following four levels,

$$\begin{tikzcd}[row sep=small]
\text{(basic) Generalized markup (SGML)} \arrow[d] \\
\text{Complete generalized markup (SGML)} \arrow[d, Rightarrow, "\cdots\cdots\cdots" left, "\cdots\cdots" right] \\
\text{Procedural (\TeX) markup} \arrow[d] \\
\text{Formatted document}
\end{tikzcd}$$

with the input phase and output (print) phase before and after. (The dots separate the SGML bound layer from the TEX bound layer.) The interfacing with procedural markup is illustrated below, with AAP's simplified table as example. This four level process resembles the coupling of higher level programming languages such as PASCAL and ADA to FORTRAN (numerical libraries). For more on the latter, I refer to Einarsson&Gentleman [1984] and other early work of mine [1984].

Note. Of course, one can also work the other way round: start from procedural marked up copy and couple that to SGML.

SGML markup, AAP's table. (Completely tagged) The complete SGML markup version — complete means expansion of short references into tags and addition of omitted (end) tags — of AAP's simple table reads:

```
<tbl>
<no>Table AAP</no>
<tt>Job Changes: 1973&ndash;1980</tt>
<hd>
   <th></th>
   <th>Gain/Loss of Hospitals
       since 1973</th>
   <th>Total No. of CEO Job Changes
       1973&ndash;80</th>
   <th>Survival Rate of CEO's</th>
</hd>
<bdy>
<row><tsb> Texas</tsb>
     <c>20</c><c></c><c>22%</c>
</row>
<row><tsb>Maryland</tsb>
     <c>5</c><c>42</c><c>24%</c>
</row>
</bdy>
<ft>
   <au>David Kinzer</au>
   <atl>Turnover Of Hospital Chief
        Executive Officers: A Hospital
        Association Perspective</atl>
   <nme>Hospital and health Services
        Administration</src>
   <dt>May&ndash;June 1982</dt>
</ft>
</tbl>
```

The above tagged table describes contents and structure. The variety of presentations for printing must be catered for with either a TEX format or a LATEX style file.

(Procedural) TEX markup, AAP's simple table. We have limited ourselves to "translating" SGML markup into procedural TEX markup (no LATEX markup). (Mainly: `<name>` into `\bname` and `</name>` into `\ename`; the transformation of the tags can be table-driven, but in the header row `\nl` commands have to be inserted manually, guided by taste and aesthetics and limited by the value of `\entrywidth`. Note that the data have to be adapted too: insertion of ? for suppressed 0, and \ before %.)

```
\entrywidth=2cm
\tablewidth=4\entrywidth
\btbl %AAP's simple example with TeX
      %procedural markup
\bno Table AAP\eno
\btt Job Changes: 1973--1980 \ett
\bhd
   \bth\eth
   \bth Gain/Loss\nl
        of Hospitals \nl
        since 1973 \eth
   \bth Total No.    \nl
        of CEO       \nl
        Job Changes  \nl
        1973--80 \eth
   \bth Survival     \nl
        Rate of      \nl
        CEO's \eth
\ehd
\btby
\brow\btsb Texas\etsb\bc$+$20\ec
     \bc\ec\bc 22\%\ec
\erow
\brow\btsb Maryland\etsb\bc$+$?5\ec
     \bc 42\ec\bc 24\%\ec
\erow
\etby
\bsrc
   \bau  David Kinzer\eau
   \batl Turnover Of Hospital Chief
   Executive Officers:
   A Hospital Association Perspective\eatl
   \bnme Hospital and health Services
   Administration\enme
   \bdt May--June 1982\edt
```

```
\esrc
\etbl
```

Note. We still have to supply the values for entrywidth and tablewidth along with each particular table, once again, manually.

TEX format macros. In order to reproduce AAP's representation the following (format) macros were written

```
%TeX ``format'' for AAP's simple table.
\newdimen\entrywidth %\entrywidth=<default>
\newdimen\tablewidth
\tablewidth=\hsize%default
\def\nl{\par\noindent}
\def\ndash{--}
%? command for nonsignificant
%  leading zeroes, Knuth p241
\catcode`?=\active
\def?{\kern\digitwidth}
%
\def\tablerule{\noalign{\hrule}}
\def\btbl{\bgroup
   \def\bno##1\eno{{\bf##1}}
   \def\btt##1\ett{{##1}\hidewidth\cr}
   \def\bhd{\tablerule\noalign{\vskip1ex}}
   \def\ehd{\cr
       \noalign{\vskip.5ex}\tablerule
       \noalign{\vskip.5ex}}
   \def\bth##1\eth{\vtop{\noindent
              \hsize=\entrywidth ##1}&}
   \def\btby{\noalign{\vskip1ex}}
   \def\etby{\noalign{\vskip1ex}}
   \def\btsb##1\etsb{\hbox to
             \entrywidth{##1\hss}\hfil}
   \def\bc##1\ec{&\hbox to .5
             \entrywidth{\hss ##1}\hfil}
   \def\brow##1\erow{##1\cr}
   \def\bsrc{\noalign\bgroup}
   \def\esrc{%Source information is
             %handled conform AAP's
             %representation
            Source: \gau, ``\gatl,''
            {\it \gnme\/}
                    \gdt.\ \gobi
            \egroup}
   % Next items are ``stored'' via gdefs
   \def\bau##1\eau{\gdef\gau{##1}}
   \def\batl##1\eatl{\gdef\gatl{##1}}
   \def\bnme##1\enme{\gdef\gnme{##1}}
   \def\bdt##1\edt{\gdef\gdt{##1}}
$$\vbox\bgroup\hsize=\tablewidth
    \halign\bgroup &##\cr%preamble line
    \tablerule\noalign{\vskip1ex}
}%end\btbl
\def\etbl{\egroup%\halign
          \egroup$$%\vbox
          \egroup%\btbl
         }%end \etbl
%end AAP simple table format
```

The above listed format macros take care of the final results in print: appropriate separators and good order and format of the 'source' items. The table entries are centered and aligned on the last digit. This required knowledge of the column width. In order to handle the footer suitably the tablewidth had to be known. The chosen approach allows flexible formatting of the footer. Variability of column widths has not been incorporated in the provided macros but can be addressed.

Difficulties with AAP's *complex* table DTD. Although not the case in the above elaborated example, we experienced difficulties with header rows which contain "halflines." In my opinion, halflines belong to the structure. Confusion arises when the `br` (begin row) and `er` (end row) attributes are used together with halflines. According to the DTD, halflines don't account for a line in the formatted result, in TEX formatting they do, jeopardizing the prescribed br- and er-values.

Note that author etc. information is stored in gdefs in TEX, in order to cope with the difference in sequence order of this items in SGML (AAP's DTD) and TEX (independent) marked up tables.

Graphics. Coupling graphics is not (yet) elaborated, because graphics in SGML is left to other sources. Various graphic sources are interfaced to SGML.[13] The only structuring aspect deals with open space (to electronically paste up the illustration) which must not be split over a page break. The difference with mathematics possibly is that formulas are also part of the text while illustrations are more or less separated from the text.

Developments

A survey of the development of SGML is given by Barron [1990].

Usage. Among the users of SGML there are:

- DOD (Automated Technical Order System)
- European Communities (FORmalised EXchange of Electronic Documents; office official publications)
- Publishers (AAP, British Library, KNUB-(Elsevier, Kluwer, ...), ...)

[13] CGM (Common Graphics Metafile) is adopted by the SGML community, as communicated by Malcolm Clark, Idle by the Thames, TeXline X.)

[21] Guittet, C.(1986): FORMEX: une mise en practique des normes internationales. SGML user's group. Bulletin, 1, 2.

[22] Herwijnen, E. van (1988): Electronic submission of Physics articles to publishers. De 1^e Nederlandse SGML conferentie. SGML: De Consequenties. (Also published in: Computer Physics Communications 57 (1989) 244 - 250: The use of text interchange standards for submitting physics articles to journals.). In the context of this paper the discussion of SGML related to TEX is relevant.)

[23] Herwijnen, E. van (priv. comm.): Streamlining publishing procedures.

[24] Herwijnen, E. van (1988): TexT Processing at CERN I. SGML Users' Group Bulletin, 3, 2, 39–54.

[25] Herwijnen, E. van (priv. comm.): Streamlining publishing procedures.

[26] Herwijnen, E. van (1990): Practical SGML. Kluwer-Academic.

[27] ISO8879 Information Processing — Text and Office Systems — Standard Generalized Markup Language (SGML). 1986–10–15.

[28] ISO/TR9573 Information Processing — SGML Support Facilities — Techniques for using SGML. 1988–09–12.

[29] Knuth, D.E. (1989): The Errors of TEX. Softw. Prac. Exp. 19, 7. 607 - 685.

[30] Kopka, H. (1989): LATEX, Eine Einführung. Addison-Wesley.

[31] Laan, C.G. van der (1984): (Graceful) Mixed Language Programming. Argonne National Laboratory Workshop.

[32] Laan, C.G. van der (1990): Typesetting Bridge via TEX. TUGboat, 11#2, 265 - 276.

[33] Laan, C.G. van der, D.C. Coleman, J.R. Luyten (1989): SGML - (La)TEX. 1. Mathematical Formulas. RC-RUG report 24.

[34] Laan, C.G. van der, D.C. Coleman (to appear): SGML - (La)TEX. 2. Tabular material.

[35] Lamport, L. (1985): LATEX a Document Preparation System. Addison-Wesley.

[36] Poppelier, N.A.F.M.(1990): SGML and TEX in Scientific Publishing. EuroTEX90, Cork.

[37] Seroul, R.(1989): Le petit livre de TEX. Inter-Éditions. Paris.

[38] MARK-IT (1989): SGML Parser, version 2. Sobemap NV, Place du Champ de Mars 5, Bte 40, 1050 Bruxelles.

[39] Schwarz, N.(1989): Einführung in TEX. Addison-Wesley. (Translated into Dutch and English)

[40] Scheller, A. (1989): Experience with SGML in the real world. Advisory Group on Computer Graphics workshop, Rutherford. (Work reported from DAPHNE-project: Document Application Processing in a Heterogeneous Network Environment.)

[41] Smith, J.M.(1987): The standard generalized markup language (SGML): Guidelines for editors and publishers. British National Bibliography Research Fund Report 26. ISBN 0-7123-3111-5.

[42] Smith, J.M.(1987): The standard generalized markup language (SGML): Guidelines for authors. British National Bibliography Research Fund Report 27. ISBN 0-7123-3112-3.

[43] SGML User's Group Newsletters. [15]

[44] Sperberg-McQueen, C.M., L. Bernard (1990): ACH-ACL-ALLC. Guidelines for the encoding and interchange of machine readable texts.

[45] Vignaud, D.(1989): L'édition structrée dans les documents, SGML applications `a l'édition française. Éditions du Cercle de la Librairie. Paris.

[46] Warmer, J., S. van Egmond (1989): The implementation of the Amsterdam SGML Parser. EP-ODD, 2, 2, 65 - 90.

[47] Warmer, J. (priv. comm).

[48] Wittbecker, A.(1989): TEX enslaved. Proceedings TUG89. Stanford. (Advantages and disadvantages of TEX-formatter with SGML "front-end" are discussed, related to DEC's VAX Document.)

[49] Wonneberger, R. (1987): Kompaktführer LATEX. Addison-Wesley.

[15] Editorial address: Pindar Infotek, 2 Grosvenor Road, Wallington, Surrey SM6 0ER, UK.