

- H. Petersen*: A  $\text{\TeX}$ -based document factory in a university environment: Processing model, implementation steps, experiences
- V. Quint, I. Vatton, H. Bedor*: GRIF: An interactive environment for  $\text{\TeX}$
- J. Röhrich*: Abstract markup in  $\text{\TeX}$
- R. Southall*: Designing a new typeface with **METAFONT**
- R. Wonneberger*: "Verheißung und Versprechen", A third generation approach to theological typesetting

### The Tiger- $\text{\TeX}$ Workstation

Susanne Lachmann  
Gesyscom GmbH, Aachen

#### Overview

The application target of the TIGER- $\text{\TeX}$  Workstation is the composition of easy to use tools for the handling of text, images and graphics in one system and the automatic integration of these different types of information into  $\text{\TeX}$ -formatted documents.

"TIGER" is the short form for "TeX-based Image- and Graphics-integration and Enhanced Representation functions".

This article describes the state of implementation, the available tools and further developments of TIGER.

#### Basic Concepts

The TIGER- $\text{\TeX}$  Workstation is the realisation of the idea of a workbench for document preparation. The set of tools is subdivided into toolboxes, each of these for the handling of a different type of information as text, image or graphic.

The tools of each toolbox are grouped dependent on the goal of actual work (i.e. editing-tools, formatting-tools, representations-tools etc.). These tool-groups correspond to different data objects which we name as manuscript, typoscript, printscript I and printscript II.

The manuscript-file contains the logically structured document content, that is the text intermixed with logical markup tags, to describe the logical structure of the document. Logical document

markup subdivides the document content into logical objects, for example headlines, chapters, paragraphs, sub-paragraphs, footnotes. The manuscript contains no information about the further layout of the document, only the definition of the logical structure and therefore the layout-specification (i.e. the formatting) is independent from any formatter. Every formatting program can be used after the logical markup tags are mapped onto fitting formatting commands and/or macros.

The typoscript-file contains the document content intermixed with formatting-commands and/or -macros. In this sense the normal  $\text{\TeX}$  source file is a typoscript.

The printscript I is the output device independent formatted state of the document, e.g. the  $\text{\TeX}$ -output DVI-file.

The printscript II contains the formatted state of the document for the output on the chosen output device, e.g. the files that are produced by the different  $\text{\TeX}$ -drivers (as VIEW, DVILaser, etc.).

#### Actual State of Implementation

The TIGER- $\text{\TeX}$  Workstation at the moment is a well-defined collection of software tools and hardware devices for the processing of each text, image and graphics with the possibility to use additional soft- and hardware within it.

#### Hardware

The TIGER- $\text{\TeX}$  Workstation is based on IBM PC XT/AT hardware with Hercules graphics card.

Printing is done on the electronic printing system ELSA (from OLYMPIA). ELSA printing is based on the xerographic process (known from modern copiers) in connection with the use of so-called magneto-optical light switching arrays (instead of the laser beam used by commonly known laser printers). The printing resolution is 300 dots per inch and the printing speed 20 pages DIN A4 (or letter size) per minute.

Other hardware devices can additionally be used within TIGER, for example a flat bed scanner or camera scanner to input pixel information (images) into the system or a OCR-Reader for text input.

### Software

The image processing toolbox of TIGER contains a pixel-editor for the creation and pixel-wise editing of image-files, -functions for the 1:1 representation on the screen and input routines to get pixel-information from above mentioned external hardware devices.

The TIGER graphics toolbox contains different CAD systems for the creation, editing and representation of vector graphics (e.g. UniCAD, AutoCAD, generic CAD).

Manuscript-preparation of texts in the above described sense of logical structuring is not yet supported by TIGER's text toolbox. For the input and editing of  $\text{\TeX}$ -sources we use an editor which is based on Borland's MS-editor including all MS-editor-functions as word wrap, block management and search and replace functions. The editor produces standard ASCII output (very fine for the  $\text{\TeX}$ -application) and can be used with Borland's Lightning spell checker. The advantage of having the editor in Turbo Pascal source also makes it possible, to create and insert any editor function you would like to have (if you're able to work with Turbo Pascal). This is very important for our work on a structure-oriented Editor for the manuscript-preparation (see further developments).

The text toolbox also contains the  $\text{PCTeX}$ -formatter, the VIEW-program and e.g. functions for printing the  $\text{\TeX}$ -source and -LOG file.

The physical integration of images and graphics into the text and the output on the electronic printing system ELSA is provided by the in-house-written driver  $\text{DVIELSA}$ . The driver gets the graphic- and image-files from their toolboxes and integrates them automatically into the text by using the special and midinsert commands of  $\text{\TeX}$ .

### Further Developments

What we plan to do and what we are actually working on is an SGML-based author's subsystem, an interactive  $\text{\TeX}$ -environment and the realization of  $\text{\TeX}$ -output on typesetters.

#### *SGML-based Author's Subsystem*

In the announced release of TIGER- $\text{\TeX}$  Workstation the described text-editor will be upgraded by an SGML-Processor and interactive functions to control and lead the author's input of document content considering a specified type of document to be created. SGML is a draft international standard of the ISO (international standardisation organization) for the definition of logical document type

structures and the description of logical structures of special documents.

Document types are for examples reports, letters, manuals or articles. The author's subsystem will have a library of available SGML-defined document types and the author will be able to add new document type definitions in an easy way. The process of creating and/or editing document-type-definitions will be controlled by an interactive user-interface, so the author of documents doesn't has to deal with SGML itself.

According to one of these document types, the manuscript editor will lead the author along the predefined logical document type structure and will control the input.

Also the author will not have to be familiar with  $\text{\TeX}$  if he doesn't want to be. The mapping of his logically structured manuscript onto a  $\text{\TeX}$ -processable typoscript will be automatically done by the system using predefined layout directives, which describe the way to replace the manuscript's logical markup tags by  $\text{\TeX}$ -macros, and this way create the typoscript. The author will have the possibility, to insert self-defined macros as layout directives, but he doesn't have to because there always will be predefined layout directives for every logical object of the document type he is working on.

One of the advantages of such an SGML-based Subsystem is the easy to perform interchanging of manuscripts to other SGML-based systems, where these manuscripts can be mapped onto different macros for other formatting programs without the need of making changes inside the manuscript itself. Another advantage is the possibility, to make  $\text{\TeX}$  useable for people, who are not interested in learning all  $\text{\TeX}$ -features, but are interested in getting fine  $\text{\TeX}$ -formatted output.

#### *Interactive $\text{\TeX}$ -Environment*

We are actually working on a  $\text{\TeX}$ -implementation in an interactive environment on a APOLLO DN300 workstation, based on the GMD-Implementation on an ICL-PERQ-System as described by W. Appelt of the GMD in a further TUGboat-release.

#### *$\text{\TeX}$ -Output on Typesetters*

In order to achieve a professional application of  $\text{\TeX}$ , we hope to finish our work on a  $\text{\TeX}$ -driver for Linotype's LTC 300/500 within this year. Great effort still has to be made to generate the  $\text{\TeX}$  TFM-files (and also the corresponding VIEW/PREVIEW pixel-files) to have the wide spectrum of Linotype's fonts available for  $\text{\TeX}$ .