WYSIWYG-TEX-EDITORS ON THE BASIS OF OBJECT-ORIENTED SYSTEM TECHNOLOGY

Eric GÖPELT, Bernd SCHMID

Cahiers GUTenberg, nº 12 (1991), p. 3-12.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1991___12_3_0>

WYSIWYG-TEX-editors on the basis of object-oriented system technology

E. GÖPELT* and B. SCHMID**

* CTB CAMTEC Software GmbH, Berlin, Federal Republic of Germany,
* Fachinformationszentrum Karlsruhe, Gesellschaft für wissenschaftlich-technische Information mbH, Eggenstein-Leopoldshafen, Federal Republic of Germany

1. WYSITEX - a TEX-editor on the basis of the object-oriented system technology

1.1. Motivation for realization of a TEX-editior

Demands of TEX-users for a change of so-far TEX-possibilities had been the first step and also motivation for the development of a WYSIWYG-capable TEX-editor.

There are to be named some in particular:

- the batch strategy for TEX-input – TEX-textinput, translation and previewing – should be overcome,
- the dualism of TEX-text and TEX-box should be offered to the user for interactive work at the same time,
- untrained TEX-workers, especially in mathmode, should get 2-dimensional construction possibilities and
- the search for syntactic and semantic mistakes should be graphical supported.

1.2. Aim of Development

The aim of development was the implementation of mathsentence as well as the partial implementation of the textsentence of plain-TEX, taking in account the demands mentioned above.

1.3. System of Development

To reach the aim, it is necessary to attach properties to each TEX-object, as, for instance, its position in text view and in box view, its superior boxes/parse nodes, its subordinate boxes/parse nodes, its mistake qualities and its modus (text/math). Thus, it is possible to select, to substitute, to highlight, to copy, to move single TEX-object.

On the basis and the circumstance that 6 months are quite a short time for realization, we choose the object oriented technology by OBJECTWORKS/SMALLTALK release 4.

1.4. User Concept

The main window consists of a text view and a box view. Edition operations in text view as well as in box view are parallel represented in the respectively other window. There are possibilities to fade in and out the text view and the box view in the main window.

Any construction views are to call over from the box view. These construction views contain pre-defined or user-defined TEX-objects (e.g.. to set up complex formula structures).

The text view offers the normal functions of a text editor, as cutting, inserting, pasting, copying and it supplies the syntactic and semantic test of the TEX-notation.

The box view and the construction views offer methods of box work, as selecting, pasting, copying, moving and cutting of box objects.

The interaction between box objects beyond window limits is possible.

As functions to interact with the host system are implemented:

- Input respectively output of TEX-documents as DVI-File by the host file system,
- Input respectively output of TEX-paragraph by the clippboard (for MS-Windows) and
- the enclosure of one of the user's previewer.

1.5. Implementation Concept

The basis for transformation of the text input stream to its box fashion was the definition of scanner objects, parser objects and box objects and the definition of the smallest work object. Then, there was to be defined their properties, the hierarchy of objects, and methods of questioning and changing of the properties.

Unit has been introduced as the smallest work object. This is either a delimiter (linefeed, tabulator, space bar), a formula (text chain limited by one dollar), a formula in display style (text chain limited by two dollars), or a text chain without dollar limitation.

A unit is a part of a unit collection, which contains all units of a TEX-document.

The reason for the introduction of the unit as the smallest work object is the limitation of effects of manipulations in text view or box views on the unit itself and at most on its predecessor and successor within the unit collection (except, of course, the cutting of whole units).

Each unit contains its scanner object tree, which reflects the structure of scanner object of one unit. Because each scanner object has the knowledge how to parse itself, and the parse result get the knowledge how it becomes a box, the unit is able to construct its parser object tree and its box object tree.

During the production process of the parse tree and the box tree each parse object gets the knowledge about its scanner object, each box object gets the knowledge about its parse object and each scanner object gets the knowledge about its box object.

Moreover, each object knows its father and sons within the respective object trees.

Thus, it is sure, that from each grip (e.g. throungh the views) all information concerning the selection object is available.

1.6. Symbol Token

Out of the mass of all scanner objects, as delimiter token, textual token or formula token array and brace token array, it should focussed on the symbol token. Symbol token as special scanner objects are particularly interesting, because they describe the whole extent of plain-TEX-command-sentence implemented by the TEX-editor.

If the scanner realize a plain-TEX command through a dictionary-equalizer within the text input stream, an adequate object, an instance of the class symbol token would be produced. This instance carries the knowledge of itself:

- the plain-TEX command as text
- the token position
- a dictionary object.

This dictionary object contains optional different information for the adequate symbol token, as

definition mode The definition mode informs about validity of the symbol token for the mathematic sentence or the text sentence or it signifies that the symbol token is valid for math and text sentence.

parser method The parser method signifies what way the symbol token produces its parse node.

box method for operators The box method for operators signifies what way the symbol token produces its box and what way its arguments are to be positioned.

box method for atomic symbol token The box method for atomic symbol token signifies what way the symbol token produce its box.

font description The font description signifies which font is to be used for the construction of the box of the symbol token

By this, the symbol token is able to fulfil each demand during its life cycle, within the TEX-editor.

Operation of the TEX-editor within the graphical user interface of the Computerized Integrated Data Base Production System (COMPINDAS) of FIZ-Karlsruhe

Beside the possibility of the stand-alone function the TEX-editor was integrated in the mask system "ObjectForms" of the companies Georg

Heeg/CTB Berlin in order to produce the graphical user interface within the project COMPINDAS of FIZ-Karlsruhe.

For that purpose, the TEX-editor was completed by the properties of a mask field.

Thus, the TEX-editor behaves like any other mask field, whereas it has two marks, first the text view with integrated syntactic and semantic tests of TEX-notation and second the box view.

Therefore, the TEX-editor was particularly able:

- to get added to other mask fields and to get relatively arranged,
- to get removed from the mask,
- to get coloured,
- to get labelled
- to get moved, enlarged and reduced on the mask by the mouse and
- to get used by application, superior to the mask system.

Eventually, the FIZ-Karlsruhe has a highly flexible, powerful and efficient WYSIWYG-user interface at his disposal, with mask and TEX fields, for the COMPINDAS project.

2. Application of a WYSIWYG-Editor in the Computerized Integrated Data Base Production System (COMPINDAS) of FIZ Karlsruhe

Fachinformationszentrum Karlsruhe is developing an integrated interactive data management system for bibliographic and factual databases called COMPINDAS. The aim of this development is to guarantee long-term database quality and efficiency of the input process.

The concept includes the following range of applications:

- Subject analysis with computer-assisted classification, indexing and translation.

- Documentation techniques with online acquisition and management of literature and factual data, recording by means of optical scanning, computer-assisted bibliographic description, control and update procedures.
- Support of the whole process by continuous surveillance of document flow.

The future development of the system will be supported by interactive programming tools.

The procedures will be performed in an integrated manner.

The system has to meet high standards for flexibility, data integrity and effectiveness of system functions. All data will be stored in one large pool independent of the type of data, the appropriate database or the subject field. To avoid duplication of work and redundancy of data storage is one main goal of the development.

The system will work online, interactive and conversational. COMPINDAS makes use of ADABAS as database management system for storage and retrieval. The applications are being generated by means of aDIS of ASTEC in Munich and a Smalltalk-80 application of Georg Heeg - Objectoriented System Technologies in Dortmund. aDIS is used for the definition of the data structures, checking routines, coupling processes, and the design of dialogue and batch routines including masks for mainframe terminals, the Smalltalk-80 application provides the graphical user interface and TEX-editing on the workstation.

Communication between mainframe and workstation will be established via Advanced Program to Program Communication (APPC, LU6.2).

2.1. Preconditions, Volume of Activities and General Aims

FIZ Karlsruhe offers a broad range of services. One of its main activities is the production of databases for all aspects of energy, physics, mathematics and related fields of science and technology. More than 230, 000 documents - articles, research reports, books, patents and conference papers - are systematically collected, analyzed and processed every year. In addition, databases comprising about 45, 000 entries with factual information are being built and maintained. They include information on conferences, research in progress, institutions, experts or products. The library catalogue of FIZ consists of 600, 000 monographic entries plus 7, 000 updates per year and 7, 000 periodicals and serials. Data are originating from very different sources and processed under various conditions, e.g. as international cooperative ventures. About 50 different products like magnetic tape services, reference journals etc. are being created using these data.

Based on many years of experience, Fachinformationszentrum is developing a new input system for the production of its bibliographic and factual databases. The precondition for the new system was that it should fit into the existing soft- and hardware surrounding. It operates on a IBM 3081 mainframe and ADABAS serves as database management system.

2.2. Requirements

COMPINDAS had to be realized as an open easy-adaptable system offering the users a maximum of flexibility and allowing them to cope with the increasing amount of data and any changes in FIZ input tasks and services.

The aim was to develop an "intelligent" documentation system based on a knowledge base including dictionaries, thesauri, rules for document analysis, formatting etc. Three major application areas had to be included:

- Subject analysis with computer-assisted classifying, indexing and translation, interactive maintenance of thesauri, dictionaries and classification schemes.
- Documentation techniques with online ordering, acquisition and management of literature and factual data, recording by means of optical scanning, computer-assisted bibliographic description, control and update procedures, creation of different and variable output products.
- Support of the whole process by continuous surveillance of document flow by means of operational statistics as well as handling and accounting service input, including maintenance, adjustment and further development of computer programs by means of interactive programming tools.

All the procedures should be performed in an integrated manner. Information must be presented homogeneously. The system is to work online, interactive and conversational. Measures and techniques to guarantee long-term database quality and efficiency of input processes are to be supported.

2.3. Concept of the System and System Components

The system will be realized in a totally interactive way. Masks are presented to the user to fill in data or update existing data. Within bibliographic and subject analysis data a broad range of special characters and mathematical formulae has to be recorded and updated as well. This has to be done using standard keyboards. This will be performed in a linearized form using $\TeX$ codes. To enable easy proofreading and final editing the codes however have to be shown in a graphical representation.

A major problem we encountered when planning our system, was how to fulfil the aim of user-friendly word processing, which in an IBM mainframe environment will always be a crucial point. We found out that by using IBM 3270 terminals we would always be restricted in one or the other way, especially in online updating. The only way out was the decision to equip COMPINDAS users with local intelligence, i.e. PC's. We could herewith also solve the problem of showing the user's formula not only in a linearized but also in a graphical form.

This brought us a new task to be fulfilled - the link of the central application on the mainframe with all the workstations/PC's running Smalltalk-80 systems. This could not be done using simple file transfer because this mechanism would have been too slow. We had to look for a mechanism linking directly a program on the IBM mainframe and a program on the PC/workstation.

In the IBM environment this can be achieved by using LU6.2 - Advanced Program to Program Communication (APPC). The mainframe system aDIS had to be extended and now includes a server part on the mainframe which uses the standard VTAM-APPC interface and a client part that runs on the PC with an interface to a standard LU6.2 software for PC's and a link to the Smalltalk-80 application.

Because Smalltalk-80 is not yet available for OS/2 (which does not support the 386-flat memory model) we had to decide to use MS-DOS with Windows 3.0 running in the extended 386-mode (Figure 1).

2.4. Demands of COMPINDAS-GUI

The graphical user interface for COMPINDAS (COMPINDAS-GUI) has to fulfil two different main requirements.

These types of requirement stem from very different types of users ranging from data typists, who have to enter great amounts of text and know by heart what is essential when using, TEX, to scientists who may not know a lot about TEX but have to construct or proofread complicated mathematical text.

The first aim is to provide the user with a flexible, easy-to- use, standardized user interface for beginners as well as experts, the second aim was to handle TEX-coded data in an efficient way with the possibility to show the result of the operations immediately and to work in a text-oriented window as well as in a graphical representation depending on the complexity of the data, the user's knowledge and his strategy.

Data being entered using this system vary extremely starting from numeric field contents and plain ASCII texts ending with text containing complicated mathematical formulae.

To record or update standard text including only few special characters, e.g. some greek letters or diacritics, a user normally does not need a graphical visualization. An online checking program will be enough. In contrary the user should not be hindered in entering text and be forced to select symbols from a menu which has to be selected first etc. - well trained data typists know such TEX codes by heart.

On the other hand it is not very easy to write down complex mathematical formulae without making any mistake. In this case it is very helpful, e.g. for the subject analyst expert, to see what he is creating. And perhaps he wants to write down things without being bothered by menus.

For proofreading and correcting these data have to be presented as in the expected printed product. Here the best way to modify a formula is not only to look at its graphical representation but also to manipulate it directly in the graphical display. This is even more desirable if the user is not a TEX expert but a scientist who is only concerned with the contents that are displayed.

We hope to offer results and experiences very soon, concerning efficiency, user acceptance and error reduction in comparison to commonly used TEX editing tools.

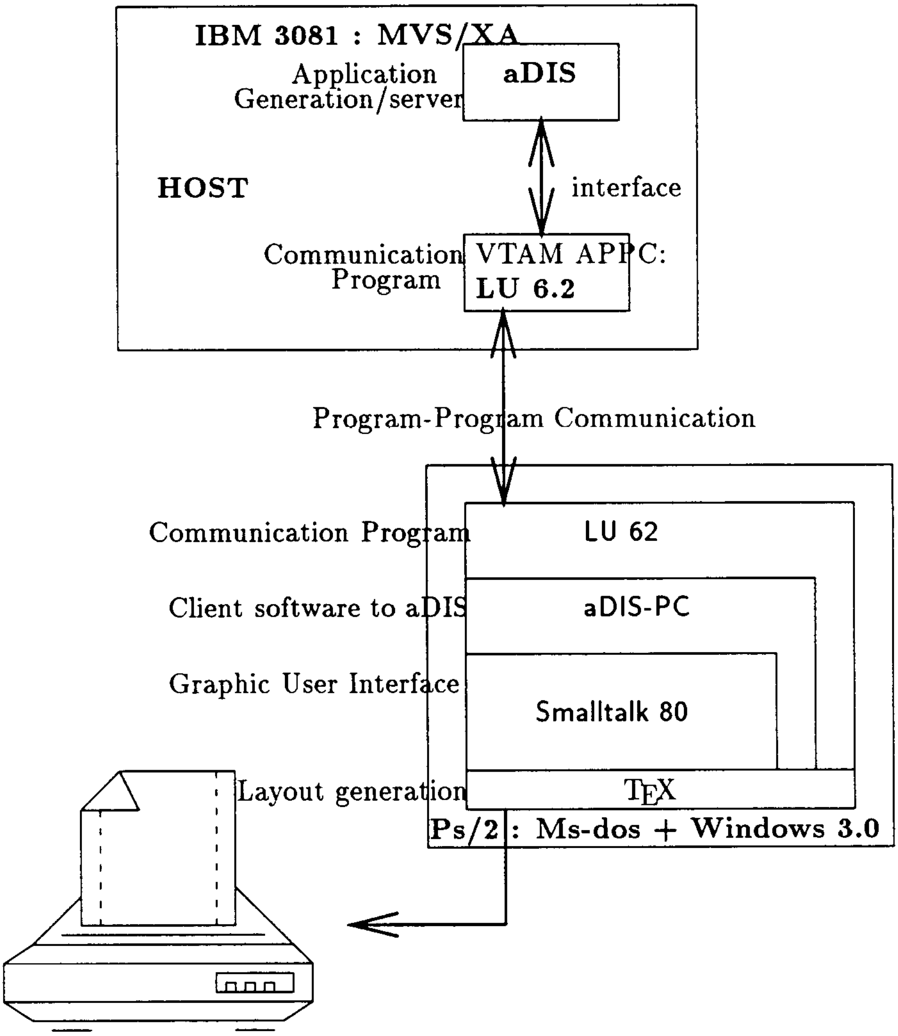

Figure 1. Software Architecture of COMPINDAS